Integration of Data Analytics and Security to enhance clinical documentation

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Information Technology

by

ROHAN CHETAN KAPOOR

16BIT0399

Under the guidance of

Prof. Praveen Kumar Reddy M.

School of Information Technology and Engineering

VIT, Vellore.

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May, 2020

DECLARATION

I hereby declare that the thesis entitled "Integration of Data Analytics and Security to enhance clinical documentation" submitted by me, for the award of the degree of *Bachelor of Technology in Information Technology* to VIT is a record of bonafide work carried out by me under the supervision of Prof. Praveen Kumar Reddy M.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Rohan

Place : Vellore

Date : 22/05/2020

Signature of the Candidate

<u>CERTIFICATE</u>

This is to certify that the thesis entitled "Integration of Data Analytics and Security to enhance clinical documentation" submitted by **Rohan Chetan Kapoor 16BIT0399**, **School of Information Technology and Engineering**, VIT, for the award of the degree of *Bachelor of Technology in Information Technology*, is a record of bonafide work carried out by him / her under my supervision during the period, 01. 12. 2018 to 30.04.2019, as per the VIT code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion meets the necessary standards for submission.

Place : Vellore

Date : 16-05-20

Signature of the Guide

Internal Examiner

External Examiner

Dr. Jasmine Norman(HoD)

Information Technology

ACKNOWLEDGEMENTS

First and foremost, I would like to thank VIT University Management for giving me the resources to carry out this project in the University. I would also like to thank our esteemed school Dean, Dr. Balakrushna Tripathy, for providing me with proper facilities for the experimentation phase of this project. I would like to thank my internal guide Prof. Praveen Kumar Reddy M, for his continued support and collaboration throughout this project. Last but not the least, I would like to thank my parents, this wouldn't have been possible without their support.

Rohan Chetan Kapoor

Executive Summary

Recent advances on prospective monitoring and retrospective analysis of health information at national or regional level are generating high expectations for the application of Big Data technologies that aim to analyze at real-time high-volumes and/or complex of data from healthcare delivery (e.g., electronic health records, laboratory and radiology information, electronic prescriptions, etc.) and citizens' lifestyles (e.g., personal health records, personal monitoring devices, social networks, etc.). Numerous reports pointed to EHR(Electronic Health Record) use as a contributor to increased physician workloads, specifically citing the time demands of clinical documentation processes. Thus Data Analytics can be employed to increase efficiency and decrease the workload from physicians. Rather than manually documenting EHRs using standard keyboard and mouse, the patient records can be processed using data analytics and visualization based on predictive modelling. The Standard approach requires physicians to document health history and physician examination data using standard tools and the rest of the visit using mining tools. The basic disadvantage of the existing system is EHR documentation places ever-increasing demands on clinicians' time, which contributes further to diminished quality of documents (example, replete with irrelevant, redundant, and erroneous information) and physician dissatisfaction. The main advantage of the proposed system is EHR documentation methods using a combination of dictation and NLP show potential for reducing documentation time and increasing usability while maintaining documentation quality, relative to EHR documentation via standard keyboard-and-mouse entry. Along these same lines, advances in the field of genomics are revolutionizing biomedical research, both in terms of data volume and prospects, as well as in terms of the social impact it entails.

TABLE OF CONTENTS

1. INTRODUCTION

1.1 OBJECTIVE

Electronic health record systems have brought instant benefit to medical organizations by reducing administrative activities, ensuring data availability, minimizing waste, enabling faster time to treatment, reducing costs and overall improving the quality of care within a health entity. The main purpose behind setting up an electronic health records is to be able to analyze voluminous, varied, unstructured health data and acquire meaningful insights through analytical and decision-making tools. Yet, while these systems have myriad advantages, they still bring up new challenges and complications to the healthcare community. Indeed, we notice the existence of different and numerous big data tools, which makes it hard to choose the right fit for a specific EHR and to exploit them largely. This paper outlines research studies, which focus on the integration of analytic models into EHR systems. The purpose is to investigate the late adoption of analytics, along with proposing a novel systematic methodology. This will allow new researchers to follow a landscape and overcome the integration issues encountered at an early stage of the adoption process

1.2 MOTIVATION

1.2.1 Electronic Health Records

Electronic health records are considered as the modern and the digital version of the health information system, which provides information on diseases, previous consultations and exam results, the EHR allows patients and healthcare professionals to store, process and share electronically medical data for the coordination of care. Through EHR systems, patient information is more easily accessible to the different departments of health care facilities for various basic health care systems. From preliminary interviews to exams, diagnostics, eventual follow- up examinations and treatment, healthcare providers can quickly have the right information in case of emergency. The blood type, allergies, diseases, possible medications or any other vital signs measurements, everything is centralized and searchable at a glance.

1.2.2 Analytics for Healthcare

Every second, dozens of terabytes of data are generated and accumulated from various sources, e.g. internet browsing, social networks, mobile transactions, online shopping and many others. Indeed, the big data paradigm has taken an expended shape, and the abundance of such structured and unstructured data has made it possible to be open to new perspectives. These new sources of data increase the chances of understanding one's behavior and motivations, identifying instant signals and triggers for someone’s interest in a specific offer or product. Getting meaningful insights from voluminous and varied amounts of data helps to understand and extract hidden information, which can be used and exploited for the proper improvement of the users’ experiences. Myriad field of studies are concerned and the healthcare sector is no exception. Indeed, the analysis of health data can assist the improvement of the quality of care for a whole population, predict new epidemics and ensure equal access to care for everyone. While health analytics is represented as one of the most important technologies in e- health, their proper deployment and integration to EHRs is not as simple as it seems.

1.2.3 Evaluating the Industry’s Integration of Analytics within EHRs

Currently, the process of implementing an EHR system is no longer a challenge for health practitioners as much as it is the case for improving analytics and gaining meaningful insights. In order to unlock the value of analytics over EHRs, we need to shed light precisely on the challenges and issues that detain their proper adoption.

1.2.4 Compatibility Issues

Introducing and studying new health IT solutions is an important thing to consider for the sake of the improvement of their capabilities. However, this can lead to crucial interoperability complications between systems. With the important expansion of the implementations of new EHR solutions, we note that the more we develop various and heterogeneous systems, the more problems of compatibility and interoperability, we encounter. While there are many EMRs, EHRs softwares and projects that are deployed in health bodies, different analytic services are also present in abundance. These latter are generally in the form of machine learning models, artificial intelligence algorithms, descriptive or predictive analytics. From the determined options, the medical experts choose the most appropriate depending on the situation. To make the right decision, experts must accurately interpret the results from tables, graphs and dashboards, which are supported by the proposed analytic solutions. Figure 1 illustrates the compatibility challenges encountered with the integration of analytics to existing EHRs. The problem raised in this sense can be expressed according to the following questions: How can we run analytics over EHR stored data? Free, paid or open source vendors? To what extant can we accept cloud analytics? How to incorporate Clinical Decision Systems (CDS) to EHRS?

The answer to the presented inquiries requires an assessment of the primary needs of the concerned health organization. As a matter of fact, health bodies should normally move towards an implementation of medical workflows rather than the adoption of separate standalone solutions. The integration of independent services of predictive, prescriptive, descriptive or diagnostic analytics induces to a poor cost and data management system, along to a complex decision support based on multiple outputs.

1.2.5 Security Measures

An approach to overcome and prevent this shoulder surfing attacks has been taken, which includes the use of vernam cipher. The Vernam Cipher is based on the principle that each plaintext character from a message is 'mixed' with one character from a key stream. If a truly random key stream is used, the result will be a truly 'random' ciphertext which bears no relation to the original plaintext. The ciphertext is generated by applying the logical XOR operation (exclusive-or) to the individual bits of plaintext and the keystream.

1.3 BACKGROUND

[1] In one of the papers, the implementation of color pass methods has been described to defend against shoulder surfing attack. Users can enter the session without revealing the actual PIN. It provides security against shoulder surfing attacks and has an intelligent user interface. However, it may take a large amount of time. The main disadvantage is that it is based on a partially observable attacker model.

[2] In one of the papers, a new method, which conceals information about the password objects as much as possible, is proposed. Besides hiding the password objects and the number of password objects, the proposed method allows both password and distracter objects to be used as the challenge set's input. The correctly entered password appears to be random and can only be derived with the knowledge of the full collection of password objects. .Therefore, it would be difficult for a shoulder-surfing adversary to identify the user's actual password. .Simulation results indicate that the correct input object and its location are random for each challenge set, .thus preventing the frequency of occurrence analysis attack. User study results show that the proposed method can avert shoulder-surfing attack. The proposed method uses images as the password, which are said to be easy to remember.

[3] In one of the papers, they have introduced a means to prevent this SSA attack. We do this through the use of advanced BW (Black White) method and session key method by changing the layout of the keypad. The simplicity would generally be a significant disadvantage of using the session key. Round redundancy, restrict to length 4, security.up to few camera-based recordings are the significant disadvantages of using a BW method.

[4] In a paper, a new scheme that uses digraph substitution rules to conceal the mechanism or activity required to derive password-images is proposed. In the proposed method, a user is only necessary to click on one of the pass-image instead of both pass-images shown in each challenge set for three consecutive sets. However, the attackers may know about the digraph substitution rules used in the proposed method.

[5] In this paper, PassBoard is a new approach to tackle the problem of the Shoulder surfing attack. Recent attempts have attempted to improvise these keyboards by dynamically loading and changing the layout to confuse the snooping miscreant.

[6] In one of the works, the user proposes an improved text-based shoulder surfing resistant graphical password scheme by using color PIN entry mechanism, which is resistant to shoulder surfing. In the proposed plan, the user can smoothly and efficiently

log in into the system. This proposed work gives security over the password from shoulder surfing and accidental login.

[7] One of the works deals with creating a mechanism specifically designed to stop the sniffing attacks from taking place. The paper.first discusses the importance of password hiding and then discusses the way to resolve the problem. .However, prevention. of sniffing using alphanumeric passwords shall be considered along with better-randomised password scrambling and manipulating operations.

[8] In one of the papers, a new hybrid graphical password-based system is proposed. The system is a combination of recognition and recall based. techniques that have many benefits as compared to the existing systems and can prove to be more convenient for the user. The scheme is resistant to shoulder surfing attacks as well as many other attacks on graphical passwords. It is proposed for smart mobile devices (like smartphones i.e. iPod, iPhone, PDAs etc), .which are more handy and convenient to use than traditional desktop computer systems.

[9] In one of the works, a new technique of user Authentication that is Graphical Password Authentication using Images Sequence, was. In this method, the user uploads images from his/her personal gallery/directory for password selection. The.images which are uploaded by one user will not be visible to another user. In this method, a graphical password is used as an alternative to textual/traditional alphanumeric passwords.

[10] One of the works introduces an early. and developing essential security elements based on Artificial Intelligence ideas, precisely, a graphical secret key with stable basic structures, which bigger things can be built based on Captcha invention of new things, and it is called Captcha as graphical passwords (CaGP). CaGP tends to different security. issues by and large, for example, guessing attacks, relay (from one place to another) attacks, and, if.joined with double view inventions of new things, bear surfing attacks. CaGP also provide a new method to handle important issues in picture hotspot graphical key.

[11] In one of the papers, they developed one system, named PassPoints, and evaluated it with human users. The outcomes of the evaluation were promising with respect to memorability of the graphical password. In the study, they expanded our human factors testing by studying two issues: firstly, the effect of tolerance, or margin of error, while clicking on the password points and secondly, the effect of the image used in the password system. In the tolerance study, the outcomes show that accurate memory for the password is sharply reduced when using a relatively small tolerance (10 x 10 pixels) around the user's password points. The outcomes show that there were little significant dissimilarity in performance of the images. This preliminary outcome suggests that several images may support memorability in graphical password systems.

[12] In another paper, a comprehensive survey of the existing graphical password techniques was conducted. Then, they provided a possible theory of their own too.

[13] A simple graphical password authentication system was proposed in one of the extended abstract. They described its operation with some examples, and highlighted significant aspects of the system.

[14] In one paper, a survey on graphical password schemes from 2005 till 2009 was presented which are proposed to be resistant against shoulder surfing attacks.

[15] This examination will concentrate on the utilization of Data Mining procedures on recently investigated informational indexes. The information mining device Weka will be utilized. Weka represents Waikato condition for information examination, and "is a mainstream suite of AI programming written in Java, created at the University of Waikato. WEKA is free programming accessible under the GNU General Public License". The reason for the investigation is to broaden past examinations by running new informational indexes of stylometry, keystroke catch, and mouse development information through Weka utilizing different information mining calculations. The examination will likewise expand past research at Pace University into the employments of a human-machine interface to build the precision of AI. To this end, the examination will utilize an ostensible informational collection, the Mushroom Database

[16] Formal Concept Analysis (FCA) is a developing information innovation that has applications in the visual investigation of huge scale information. Be that as it may, informational indexes are regularly excessively huge (or contain such a large number of formal ideas) for the subsequent idea grid to be decipherable. This paper supplements existing work here by portraying two techniques by which valuable and sensible grids can be gotten from substantial informational collections. This is accomplished however the utilization of a lot of uninhibitedly accessible FCA apparatuses: the setting maker FcaBedrock and the idea excavator In-Close, that were created by the creators, and the cross section manufacturer ConExp. In the primary strategy, a sub-setting is created from an informational collection, offering ascend to a discernible grid that centers around qualities of intrigue. In the second strategy, a setting is dug for 'extensive' ideas which are then used to re-compose the first setting, in this manner diminishing 'clamor' in the unique circumstance and offering ascend to an intelligible cross section that clearly depicts a reasonable diagram of the substantial arrangement of information it is gotten from.

[17] This paper presents grouping procedures for breaking down mushroom dataset. Fake Mushroom dataset is made out of records of various kinds of mushrooms, which are palatable or non-eatable. Aritificial Neural Network and Adaptive Nuero Fuzzy deduction framework are utilized for execution of the grouping strategies. Diverse strategies utilized for grouping like ANN, ANFIS and Naïve Bayes are utilized to order extraordinary mushrooms as consumable or non-palatable. The execution of the distinctive systems is assessed utilizing exactness, MAE, kappa measurement. In the wake of examining the outcomes it was discovered that Adaptive Nuero Fuzzy derivation System outflanked different strategies with most elevated exactness, least mean outright mistake and ANN is the second best entertainer. In the event that size of preparing set is expanded, the precision additionally expanded as for preparing set.

[18] Information mining assumes an essential job in our every day life period. Every one of the information has been digitalized so we have to break down it to make helpful data for our insight. Order and Clustering are the two essential real methods utilized for extricating the information from the database. Bunching is known as the

unsupervised realizing which is segment a dataset in to a gathering by their similitudes. The goal of this paper is to assess the execution of various grouping calculation, for example, Expectation Maximization (EM), Farthest Fast and K-implies by accurately bunched occurrences and time taken to construct the model for mushroom dataset utilizing information mining instrument WEKA (Waikato condition for Knowledge Analysis). The mushroom dataset comprises of 8124 occurrences and 22 properties with two classes whether it is palatable or harmful. The dataset is gathered from the UCI AI archive.

[19] In this paper, an information digging application is presented for choosing exceedingly powerful factors or side effect of various infection finding in mushroom yield. The investigation additionally centers around a few variables causing a particular mushroom infection. Exceedingly potential manifestations among a few variables were engaged out for better administration in such manner. That is the reason information mining systems are being utilized for positioning among side effects. This paper centers around recognizing explicit illnesses among a few infections utilizing an information mining grouping based methodologies. Genuine information has been taken from mushroom ranch and from there on sanitization of potential elements is done through information mining approaches. The arrangement method and sickness forecast of mushroom dataset were readied utilizing Naïve Bayes, SMO and RIDOR calculations. A measurable examination has been created so as to locate the best indications required for mushroom sickness analysis. Other than this, it looks through the best performing arrangement calculation among all.

[20] Extraction of information in country information is a trying undertaking, from discovering plans likewise, associations additionally, elucidation. In solicitation to procure possibly interesting structures likewise, associations from this information, it is consequently imperative that a procedure be made additionally, exploit the arrangements of existing techniques likewise, instruments open for information mining additionally, information very in databases. Information mining is moderately another methodology in the field of horticulture. Exact information in depicting crops relies upon climatic, geological, common likewise, different components. These are exceptionally basic contributions to make depiction likewise, desire models in

information mining. In this examination, a powerful information mining system dependent on kNN is investigated, presented additionally, executed to depict country crops. The methodology attracts moves up to demand issues by using Principal Components Analysis (kNN) as a pre getting ready technique additionally, a changed Genetic Algorithm (GA) as the limit streamlining agent. The health limit in GA is changed appropriately using powerful partition measure.

2. PROJECT DESCRIPTION AND GOALS

2.1 Dataset Description and Sample Data

This dataset provides data on foodborne disease outbreaks reported to CDC from 1998 through 2015. Data fields include year, state (outbreaks occurring in more than one state are listed as "multistate"), location where the food was prepared, reported food vehicle and contaminated ingredient, etiology (the pathogen, toxin, or chemical that caused the illnesses), status (whether the etiology was confirmed or suspected), total illnesses, hospitalizations, and fatalities. In many outbreak investigations, a specific food vehicle is not identified; for these outbreaks, the food vehicle variable is blank.

Next time you take a bite, consider this: roughly one in six (or 48 million) people in the United States get sick from eating contaminated food per year. More than 250 pathogens and toxins have been known to cause foodborne illness and almost all of them can cause an outbreak.

A foodborne disease outbreak occurs when two or more people get the same illness from the same contaminated food or drink. While most foodborne illnesses are not part of a recognized outbreak, outbreaks provide important information on how germs spread, which foods cause illness, and how to prevent infection.

Public health agencies in all 50 states, the District of Columbia, U.S. territories, and Freely Associated States have primary responsibility for identifying and investigating outbreaks and use a standard form to report outbreaks voluntarily to CDC. During 1998–2008, reporting was made through the electronic Foodborne Outbreak Reporting System (eFORS).

```
<class 'pandas.core.frame.DataFrame'>
```

```
Int64Index: 19119 entries, 1998 to 2015

Data columns (total 11 columns):

 #   Column             Non-Null Count  Dtype

---  ------             --------------  -----

 0   Month              19119 non-null  object

 1   State              19119 non-null  object

 2   Location           16953 non-null  object

 3   Food               10156 non-null  object

 4   Ingredient         1876 non-null   object

 5   Species            12500 non-null  object

 6   Serotype/Genotype  3907 non-null   object

 7   Status             12500 non-null  object

 8   Illnesses          19119 non-null  int64

 9   Hospitalizations   15494 non-null  float64

 10  Fatalities         15518 non-null  float64

dtypes: float64(2), int64(1), object(8)

memory usage: 1.8+ MB
```

2.2 Functional Process flow with Schematic

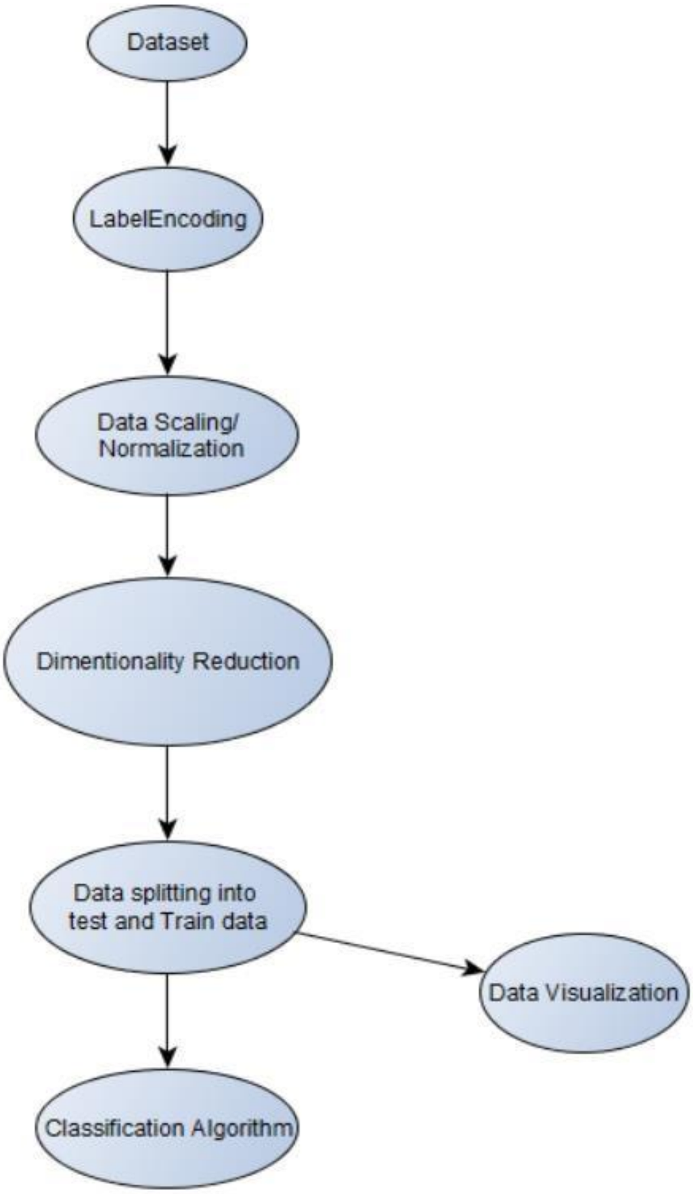


Fig 2.2.1: Flowchart of Project Workflow

The aforementioned flowchart is followed by a security module. Initially, the user is prompted to give two inputs one being the registration email and the other being the password, which the user may choose from the given list of suggested passwords, which are tested for being difficult and highly inconvenient to crack or may define a password explicitly. Once the other pre-requisite details are entered, a secret keyboard will pop up and the user will be using it to enter the password securely. When user enters a character key, the software generates a unique keyboard with a new sequence by using Vernam Cipher. .The plaintext is combined with a random stream of data of the same length in order to generate the cipher text by using XOR function. This procedure is unique for every session and every user which makes it highly secure.

3. TECHNICAL SPECIFICATIONS

In this project, the main language used is python. In addition, I use the following Dependencies to achieve the algorithm.

i. NumPy
ii. SciPy
iii. Scikit
iv. matplotlib
v. sklearn
vi. Keras
vii. Seaborn
viii. pandas

3.1 Languages Involved

3.1.1 Python – It is a general purpose and high level programming language. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks.

3.1.2 HTML, CSS ,JS and pHp – This is instrumental mainly in the security phase for designing an interactive keyboard to enhance the security of potentially sensitive information against camera based attacks.

3.2 Softwares Involved

Theano Backend is deployed on a Jupyter environment. Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.
Besides that Windows 10 OS is used for the processing along with Google chrome for displaying the various documentation.

4. DESIGN APPROACH AND DETAILS

4.1 Methods Involved

4.1.1 Data Understanding

First, this phase starts with the collection of data. To perform data collection, there are activities that need to be performed, such as data load and data integration. Next, the “gross” or “surface” properties of the acquired data need to be examined and reported. Then, we need to explore the data needs by tackling the data mining questions. That can be addressed using querying, reporting, and visualization. Finally, we have to examine the data quality by answering some important questions, such as:“Is there any missing values in the acquired data?”

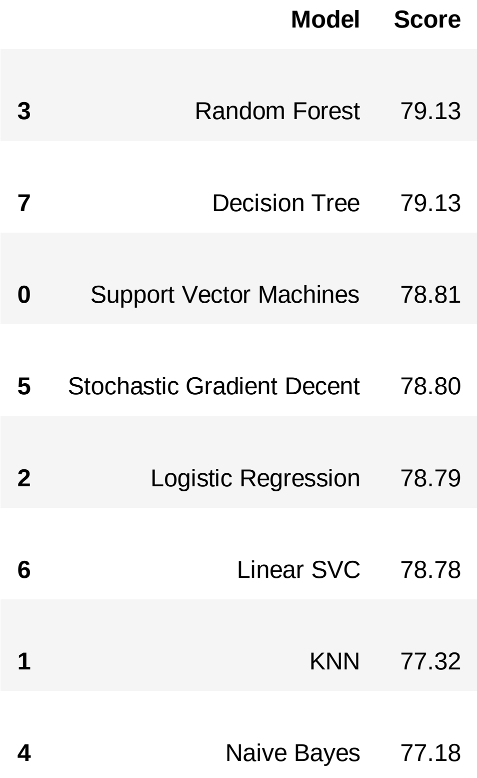
4.1.2 Data Modelling

In this process, we have applied methods to extract patterns from the data. Also, this mining includes several tasks, such as classification, prediction, clustering, time series analysis, and so on. Pattern evaluation identifies the truly interesting patterns that represent knowledge based on different types of interesting measures. A pattern is considered to be interesting if it is potentially useful and easily understandable. Further, it validates some hypothesis that someone wants to confirm new data with some degree of certainty.

4.1.3 Data Pre-processing

In the data mining process, data gets cleaned, as data in the real world is noisy, inconsistent, and incomplete. Data cleaning includes a number of techniques, such as filling in the missing values and combined compute. In this process, data in integrated from different data sources, as data is in different formats in different locations. We can store data in a database, text files, spreadsheets, documents, data cubes, and so on. Although, data integration is complex because normally data doesn’t match the different sources.

Table 4.1: Comparison of various classifiers



	Model	Score
3	Random Forest	79.13
7	Decision Tree	79.13
0	Support Vector Machines	78.81
5	Stochastic Gradient Decent	78.80
2	Logistic Regression	78.79
6	Linear SVC	78.78
1	KNN	77.32
4	Naive Bayes	77.18

4.1.3 Data Analytics

In this process, we have to transform and consolidate the data into different forms that's suitable for mining. Normally this process includes normalization, aggregation, generalization, etc. This is the process by which data relevant to the analysis is retrieved from the database. This process requires large volumes of historical data for analysis, as usually the data repository with integrated data contains much more data than actually required. From the available data, data of interest needs to be selected and stored.

4.1.5 Security Module

The Vernam Cipher is based on the theory that every plaintext character from a message is 'mixed' with one character of a key stream. If an absolutely random key stream is used, the outcome will be a truly 'random' cipher text which bears no relation or resemblance to the original plaintext. In that case, the cipher is quite similar to the unbreakable One-Time Pad (OTP). As it was usually used with teleprinters and 5-level punched tape, the system is also called One-Time Tape or OTT.

If the resulting cipher text in the OTT system described above is in fact absolutely random, it can securely be sent over the air, with no the risk of being deciphered by an eavesdropper. All the recipient should do is combining the cipher text with the same OTT to reveal the original plaintext. One only has to assure that the OTT is so random, that there are only two copies of it and that both of these copies are destroyed immediately after their usage and that they are used only once.

The generation of the cipher text is done by applying the logical XOR operation (exclusive-or) to the individual bits of plaintext as well as the key stream.

plaintext + key = cipher text ⇒ cipher text + key = plaintext . . . .

In mathematics, the XOR operation is called modulo-2 addition. In this case, each bit of the plaintext is XOR-ed with each bit of the key. The resultant bit will only be '1' if the two input bits are different from each other. If they are equal (both 1 or both 0), the outcome will be '0'. For example, take the letter 'A', which is represented by 00011, and add it to the letter 'B', represented by 11001.

Initially, the user is prompted to give two inputs one being the registration email and the other being the password which the user may choose from the given list of suggested passwords which are tested for being difficult and highly inconvenient to crack or may define a password explicitly. Once the other pre-requisite details are entered, a secret keyboard will pop up and the user will be using it to enter the password securely. When user enters a character key, the software generates a unique keyboard with a new sequence by using Vernam Cipher. .The plaintext is combined with a random stream of data of the same length in order to generate the cipher text by using XOR function. This procedure is unique for every session and every user which makes it highly secure.

4.2 Codes and Standards

4.2.1 Data Analytics and Data Pre-Processing

```
import numpy
import matplotlib.pyplot as plt
import seaborn as sns
import pandas
%matplotlib inline
plt.rcParams['figure.figsize'] = (16.0, 4.0)
sns.set_style("whitegrid")
numpy.random.seed(7)
                                                                   In [2]:
data = pandas.read_csv('D:\Project\outbreaks.csv', index_col=[0])
data.head()
                                                                   Out[2]:
                                                                  In [39]:
plt.figure( figsize = (25,50))
sns.countplot('State', data = data_u)
plt.show()


plt.show()

                                                                   In [9]:
import os
os.environ['KERAS_BACKEND']='theano'
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn import model_selection
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, roc_auc_score, matthew
s_corrcoef
from sklearn.metrics import classification_report




import pandas as pd
df_Location = pd.get_dummies(data['Location'])
```

```
df_Food = pd.get_dummies(data['Food'])
df_Species = pd.get_dummies(data['Species'])
df_Serotype = pd.get_dummies(data['Serotype/Genotype'])
df_Status = pd.get_dummies(data['Status'])
df_concat = pd.concat([data, df_Location, df_Food, df_Species, df_Serotype, df_Status ], axis=1)
print (df_concat.head())
```

In [11]:

```
df_concat.head(20)
```

Out [20 rows × 3761 columns

In [12]:

```
df_concat.drop(['Location', 'Food','Species','Serotype/Genotype','Status', 'Suspected'], inplace=True, axis=1)
print (df_concat.head())
```

In [13]
:

```
df_concat.fillna(0)
```

In [17]:

```
df_concat.drop(df_concat.iloc[:, 6:3755], inplace = True, axis = 1)
```

In [18]:

```
df_concat.head(30)
```

Out[18]:

In [20]:

```
#plots a bar graph showing food bourne disease are not deadly as their are minimal fatalities
plt.figure( figsize = (25,15))
sns.countplot('Fatalities', data = df_concat)
plt.show()
```

In [21]:

```
df_concat.drop(['Ingredient','Month','State'], axis = 1, inplace = True)
```

In [22]:

```
df = df_concat.fillna(0)
df.head(30)
```

Out[22]:

In [23]:

```
from numpy import unique
from numpy import where
from sklearn.datasets import make_classification
from sklearn.cluster import AffinityPropagation
from matplotlib import pyplot
X, _ = make_classification(n_samples=1000, n_features=2, n_informative=2, n_redundant=0, n_clusters_per_class=1, random_state=4)
# define the model
model = AffinityPropagation(damping=0.9)
# fit the model
model.fit(X)
```

```
# assign a cluster to each example
yhat = model.predict(X)
# retrieve unique clusters
clusters = unique(yhat)
# create scatter plot for samples from each cluster
for cluster in clusters:
        # get row indexes for samples with this cluster
        row_ix = where(yhat == cluster)
        # create scatter of these samples
        pyplot.scatter(X[row_ix, 0], X[row_ix, 1])
# show the plot
pyplot.show()
```

In [24]:

```
plt.figure(figsize=(5,1))
sns.heatmap(df.corr(), annot=True, cmap='seismic_r', linewidths=.5)
plt.show()
```

In [25]:

```
df.drop(['Fatalities'], axis = 1, inplace = True)
# split into input (X) and output (Y) variables
X = df.iloc[:,0:1].astype(float)
Y = df.iloc[:,-1]

import numpy
import matplotlib.pyplot as plt
import seaborn as sns
import pandas
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0, 1))
X = scaler.fit_transform(X)
```

In [26]:

```
test_size = 0.1
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(
X, Y, test_size=test_size, random_state=7)
```

In [29]:

```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC, LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.linear_model import Perceptron
from sklearn.linear_model import SGDClassifier
from sklearn.tree import DecisionTreeClassifier
random_forest = RandomForestClassifier(n_estimators=100)
random_forest.fit(X_train, Y_train)
ran_pred = random_forest.predict(X_test)
random_forest.score(X_train, Y_train)
```

```
acc_random_forest = round(random_forest.score(X_train, Y_train) * 10
0, 2)
acc_random_forest
```

In [30]:

```
logreg = LogisticRegression()
logreg.fit(X_train, Y_train)
log_pred = logreg.predict(X_test)
acc_log = round(logreg.score(X_train, Y_train) * 100, 2)
acc_log
```

In [31]:

```
svc = SVC()
svc.fit(X_train, Y_train)
svc_pred = svc.predict(X_test)
acc_svc = round(svc.score(X_train, Y_train) * 100, 2)
acc_svc
```

Out[31]:

```
78.81
```

In [32]:

```
knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
knn_pred = knn.predict(X_test)
acc_knn = round(knn.score(X_train, Y_train) * 100, 2)
acc_knn
```

Out[32]:

```
77.32
```

In [33]:

```
gaussian = GaussianNB()
gaussian.fit(X_train, Y_train)
gnb_pred = gaussian.predict(X_test)
acc_gaussian = round(gaussian.score(X_train, Y_train) * 100, 2)
acc_gaussian
```

Out[33]:

```
77.18
```

In [34]:

```
linear_svc = LinearSVC()
linear_svc.fit(X_train, Y_train)
lin_pred = linear_svc.predict(X_test)
acc_linear_svc = round(linear_svc.score(X_train, Y_train) * 100, 2)
acc_linear_svc
```

Out[34]:

```
78.78
```

In [35]:

```
sgd = SGDClassifier()
sgd.fit(X_train, Y_train)
sgd_pred = sgd.predict(X_test)
acc_sgd = round(sgd.score(X_train, Y_train) * 100, 2)
acc_sgd
```

Out[35]:

```
78.8
```

In [36]:

```
decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
dec_pred = decision_tree.predict(X_test)
acc_decision_tree = round(decision_tree.score(X_train, Y_train) * 10
0, 2)
acc_decision_tree
```

Out[36]:

```
79.13
```

In [37]:

```
models = pd.DataFrame({
    'Model': ['Support Vector Machines', 'KNN', 'Logistic Regression
',
              'Random Forest', 'Naive Bayes',
              'Stochastic Gradient Decent', 'Linear SVC',
              'Decision Tree'],
    'Score': [acc_svc, acc_knn, acc_log,
              acc_random_forest, acc_gaussian,
              acc_sgd, acc_linear_svc, acc_decision_tree]})
models.sort_values(by='Score', ascending=False)
```

Out[37]:

```
model = SVC()
model = KNeighborsClassifier()
model = GaussianNB()
model = LinearDiscriminantAnalysis()
model = LogisticRegression()
model.fit(X_train, Y_train)
predicted = model.predict(X_test)
matrix = confusion_matrix(Y_test, predicted)
print(matrix)
from sklearn.metrics import classification_report, matthews_corrcoef
, accuracy_score

report = classification_report(Y_test, predicted)
```

[33]:

```
from sklearn.model_selection import cross_val_predict, cross_val_sco
re
from sklearn.metrics import confusion_matrix,classification_report,a
ccuracy_score
```

In [34]:

```
def print_score(classifier,X_train,Y_train,X_test,Y_test,train=True)
:
    if train == True:
        print("Training results:\n")
        print('Accuracy Score: {0:.4f}\n'.format(accuracy_score(Y_tr
ain,classifier.predict(X_train))))
```

```
        print('Classification Report:\n{}\n'.format(classification_r
eport(Y_train,classifier.predict(X_train))))
        print('Confusion Matrix:\n{}\n'.format(confusion_matrix(Y_tr
ain,classifier.predict(X_train))))
        res = cross_val_score(classifier, X_train, Y_train, cv=10, n
_jobs=-1, scoring='accuracy')
        print('Average Accuracy:\t{0:.4f}\n'.format(res.mean()))
        print('Standard Deviation:\t{0:.4f}'.format(res.std()))
    elif train == False:
        print("Test results:\n")
        print('Accuracy Score: {0:.4f}\n'.format(accuracy_score(Y_te
st,classifier.predict(X_test))))
        print('Classification Report:\n{}\n'.format(classification_r
eport(Y_test,classifier.predict(X_test))))
        print('Confusion Matrix:\n{}\n'.format(confusion_matrix(Y_te
st,classifier.predict(X_test))))
```

In [35]:

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression()

classifier.fit(X_train,Y_train)
```

Out[35]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercep
t=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, ver
bose=0,
                   warm_start=False)
```

In [36]:

```
print_score(classifier,X_train,Y_train,X_test,Y_test,train=True)
```

```
Training results:

Accuracy Score: 0.7904

Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      1.00      0.88     10579
         1.0       0.00      0.00      0.00      1290
         2.0       0.00      0.00      0.00       582
         3.0       0.00      0.00      0.00       291
         4.0       0.00      0.00      0.00       164
         5.0       0.00      0.00      0.00       115
         6.0       0.00      0.00      0.00        79
         7.0       0.00      0.00      0.00        49
         8.0       0.00      0.00      0.00        42
         9.0       0.00      0.00      0.00        24
        10.0       0.00      0.00      0.00        20

    accuracy                           0.79     13383
   macro avg       0.01      0.02      0.02     13383
weighted avg       0.62      0.79      0.70     13383


Confusion Matrix:
[[10578     0     1 ...     0     0     0]
 [ 1290     0     0 ...     0     0     0]
 [  582     0     0 ...     0     0     0]]

Average Accuracy:      0.7904

Standard Deviation:    0.0004
```

In [37]:

```
print_score(classifier,X_train,Y_train,X_test,Y_test,train=False)
```

```
Test results:

Accuracy Score: 0.7821

Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      1.00      0.88      4485
         1.0       1.00      0.00      0.00       596
         2.0       0.00      0.00      0.00       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16


    accuracy                           0.78      5736
   macro avg       0.05      0.03      0.02      5736
weighted avg       0.72      0.78      0.69      5736


Confusion Matrix:
[[4485    0    0 ...    0    0    0]
[ 595     1     0 ...     0     0     0]
[ 262     0     0 ...     0     0     0]]
```

4.2.1 Data modelling

```
from math import floor
def partition(vector, fold, k):

    size = vector.shape[0]

    start = floor((size/k)*fold)

    end = floor((size/k)*(fold+1))

    validation = vector[start:end]

    #print(str(type(vector)))

    if str(type(vector)) == "<class 'scipy.sparse.csr.csr_matrix'>":

        indices = range(start, end)

        mask = numpy.ones(vector.shape[0], dtype=bool)

        mask[indices] = False

        training = vector[mask]

    elif str(type(vector)) == "<class 'numpy.ndarray'>":

        training = numpy.concatenate((vector[:start], vector[end:]))

    return training, validation



def Cross_Validation(learner, k, examples, labels):

    train_folds_score = []

    validation_folds_score = []

    test_score_auc = []

    test_score_mcc = []
    for fold in range(0, k):

        training_set, validation_set = partition(examples, fold, k)

        training_labels, validation_labels = partition(labels, fold,
 k)

        learner.fit(training_set, training_labels)

        training_predicted = learner.predict(training_set)

        validation_predicted = learner.predict(validation_set)

        test_predicted = learner.predict(X_test)

        train_folds_score.append(roc_auc_score(training_labels, trai
ning_predicted))

        validation_folds_score.append(roc_auc_score(validation_label
s, validation_predicted))

        test_score_auc.append(roc_auc_score(Y_test, test_predicted))

        test_score_mcc.append(matthews_corrcoef(Y_test, test_predict
ed))

    return train_folds_score, validation_folds_score, test_score_auc
, test_score_mcc
```

In [23]:

```
def run(model) :
    train_scores, validation_scores, test_scores_auc, test_scores_mc
c = Cross_Validation(model, 10, X_test, Y_test)
    #print(train_scores, validation_scores, test_scores)
    print(model)
    print('Train AUC', float(format(numpy.mean(train_scores), '.3f')
))
    print('Validation AUC',float(format(numpy.mean(validation_scores
), '.3f')))
    print('Test AUC',float(format(numpy.mean(test_scores_auc), '.3f'
)))
    print('Test MCC',float(format(numpy.mean(test_scores_mcc), '.3f'
)))
    print()
```

In [25]:

```
model = Sequential()
model.add(Dense(256, input_dim=1, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

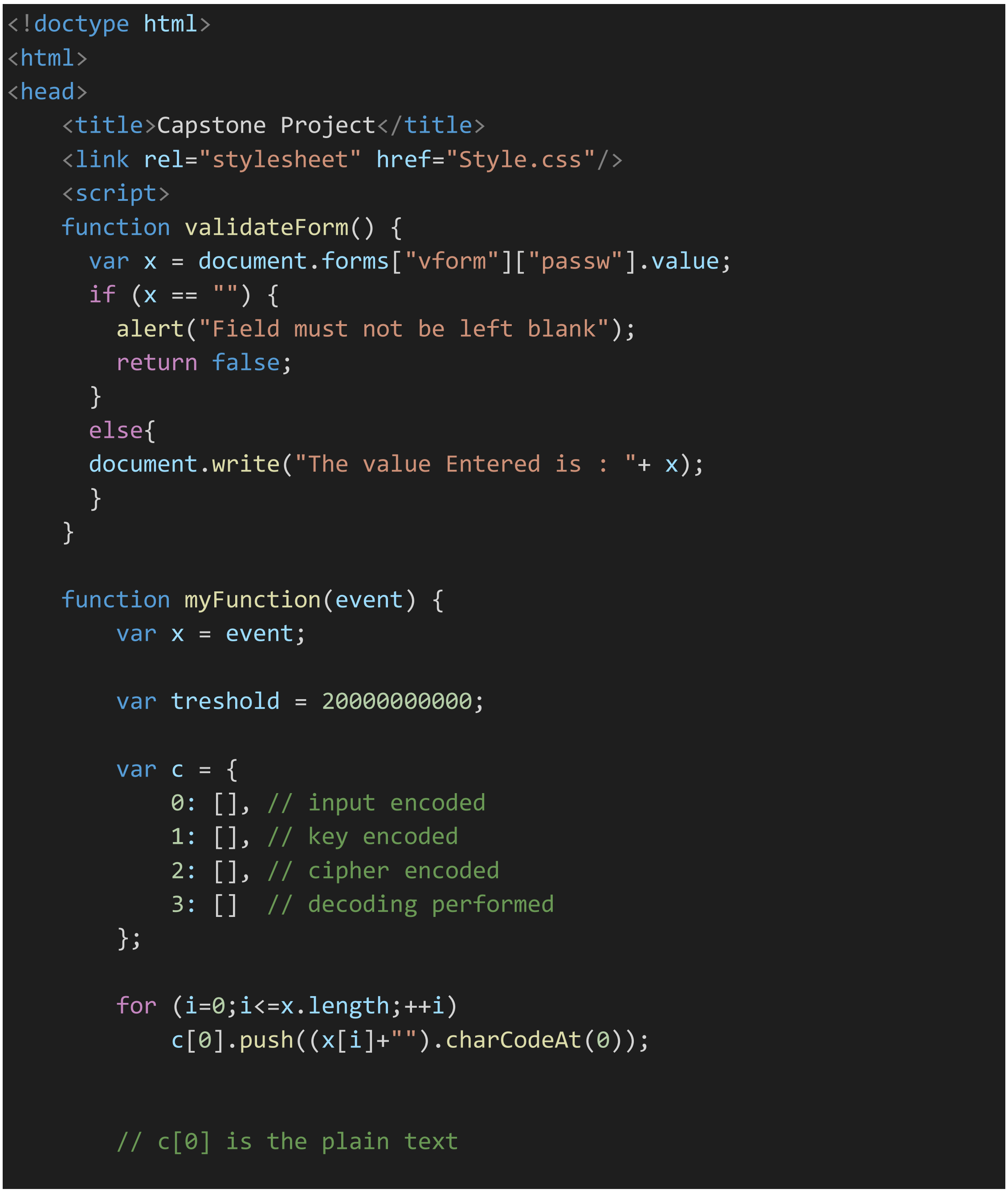
In [ ]:

```
history = model.fit(X_train, Y_train, epochs=100, batch_size=4, verb
ose=1, validation_split=0.1)
```

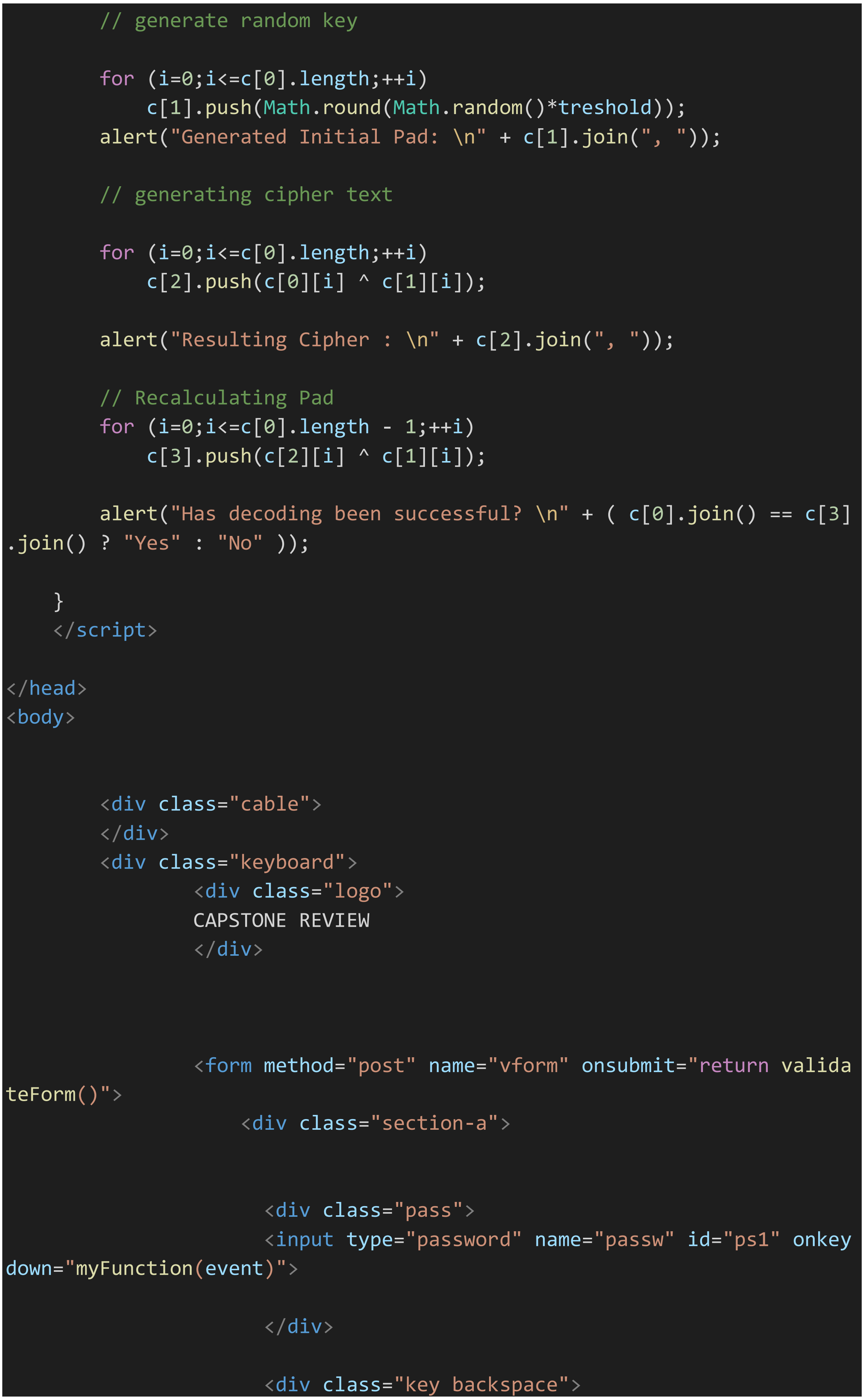
```
Train on 15486 samples, validate on 1721 samples
Epoch 1/100
15486/15486 [==============================] - 55912s 4s/step - loss:
 0.1537 - accuracy: 0.1620 - val_loss: 0.2512 - val_accuracy: 0.0924
Epoch 2/100
15486/15486 [==============================] - 20376s 1s/step - loss:
 -0.9542 - accuracy: 0.2514 - val_loss: -0.9563 - val_accuracy: 0.579
3
Epoch 3/100
15486/15486 [==============================] - 22259s 1s/step - loss:
 -2.5810 - accuracy: 0.3733 - val_loss: -0.8726 - val_accuracy: 0.092
4
Epoch 4/100
15486/15486 [==============================] - 19561s 1s/step - loss:
 -3.0549 - accuracy: 0.4407 - val_loss: -1.1454 - val_accuracy: 0.092
4
Epoch 5/100
15486/15486 [==============================] - 24930s 2s/step - loss:
 -3.2108 - accuracy: 0.4725 - val_loss: -1.4446 - val_accuracy: 0.567
7
Epoch 6/100
15486/15486 [==============================] - 20879s 1s/step - loss:
 -3.3477 - accuracy: 0.5033 - val_loss: -1.3890 - val_accuracy: 0.616
5
Epoch 7/100
15486/15486 [==============================] - 20974s 1s/step - loss:
 -3.3401 - accuracy: 0.5138 - val_loss: -1.3531 - val_accuracy: 0.645
0
Epoch 8/100
15486/15486 [==============================] - 19218s 1s/step - loss:
 -3.3214 - accuracy: 0.5265 - val_loss: -1.4889 - val_accuracy: 0.549
1
Epoch 9/100
15486/15486 [==============================] - 14305s 924ms/step - lo
ss: -3.2661 - accuracy: 0.5303 - val_loss: -1.4542 - val_accuracy: 0.
6049
Epoch 10/100
```

```
15486/15486 [==============================] - 21324s 1s/step - loss:
 -3.4066 - accuracy: 0.5426 - val_loss: -1.3945 - val_accuracy: 0.637
4
Epoch 11/100
15486/15486 [==============================] - 23875s 2s/step - loss:
 -3.4112 - accuracy: 0.5447 - val_loss: -1.4849 - val_accuracy: 0.579
3
```

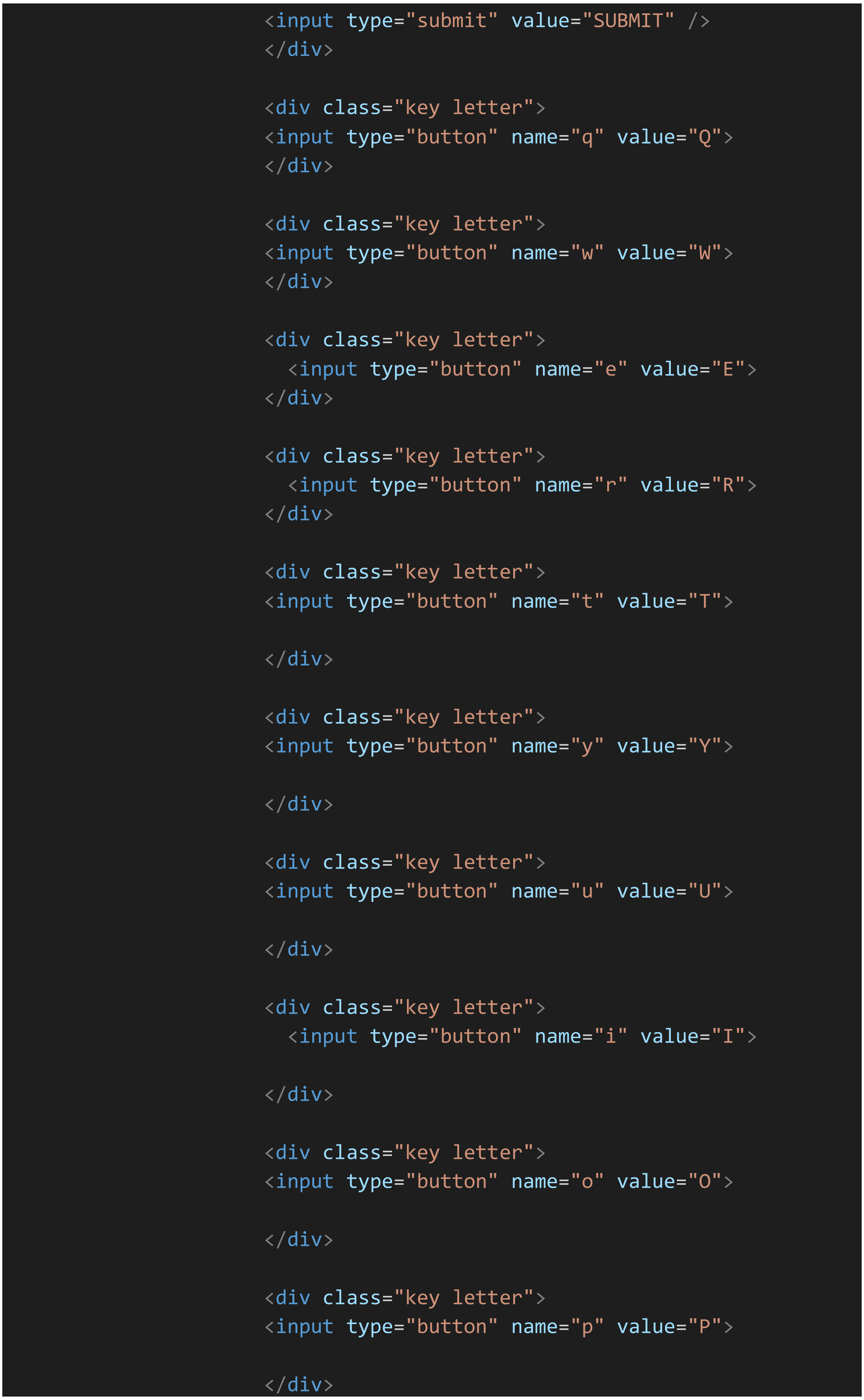
4.2.3 Security Module



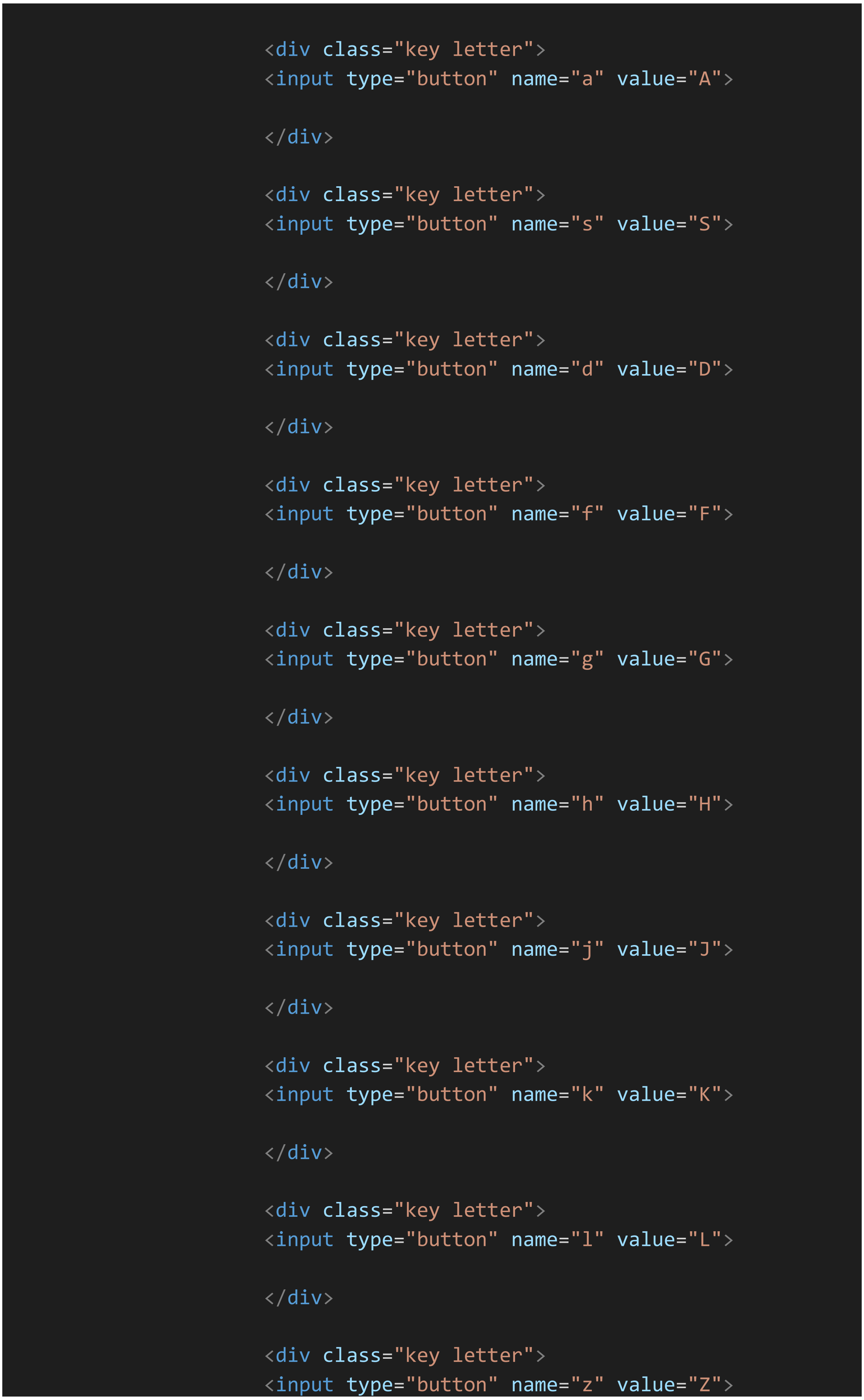
```
<!doctype html>
<html>
<head>
    <title>Capstone Project</title>
    <link rel="stylesheet" href="Style.css"/>
    <script>
    function validateForm() {
      var x = document.forms["vform"]["passw"].value;
      if (x == "") {
        alert("Field must not be left blank");
        return false;
      }
      else{
      document.write("The value Entered is : "+ x);
      }
    }

    function myFunction(event) {
        var x = event;

        var treshold = 20000000000;

        var c = {
            0: [], // input encoded
            1: [], // key encoded
            2: [], // cipher encoded
            3: []  // decoding performed
        };

        for (i=0;i<=x.length;++i)
            c[0].push((x[i]+"").charCodeAt(0));


        // c[0] is the plain text
```



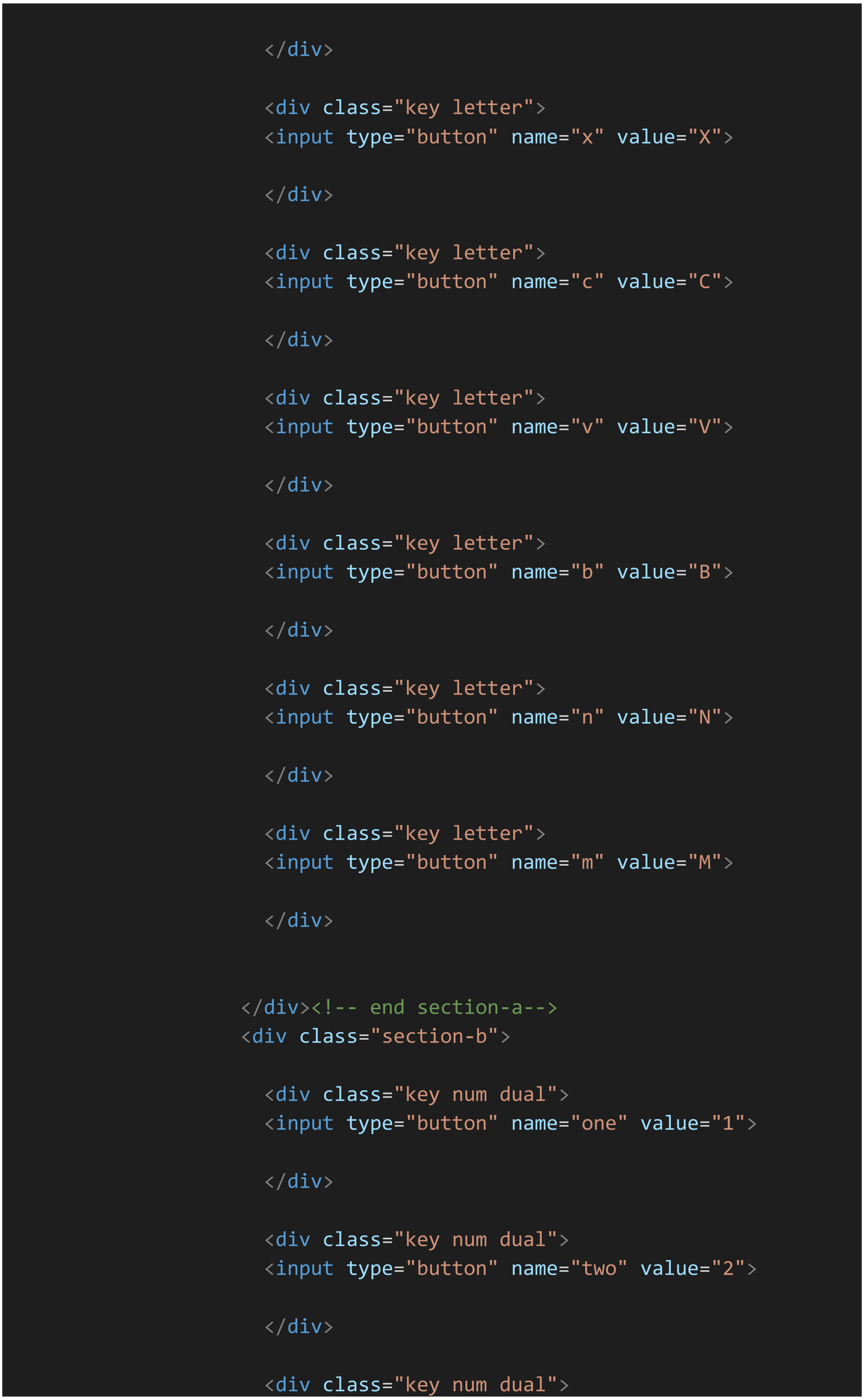
```
        // generate random key

        for (i=0;i<=c[0].length;++i)
            c[1].push(Math.round(Math.random()*treshold));
        alert("Generated Initial Pad: \n" + c[1].join(", "));

        // generating cipher text

        for (i=0;i<=c[0].length;++i)
            c[2].push(c[0][i] ^ c[1][i]);

        alert("Resulting Cipher : \n" + c[2].join(", "));

        // Recalculating Pad
        for (i=0;i<=c[0].length - 1;++i)
            c[3].push(c[2][i] ^ c[1][i]);

        alert("Has decoding been successful? \n" + ( c[0].join() == c[3]
.join() ? "Yes" : "No" ));

    }
    </script>

</head>
<body>


        <div class="cable">
        </div>
        <div class="keyboard">
                <div class="logo">
                CAPSTONE REVIEW
                </div>




                <form method="post" name="vform" onsubmit="return valida
teForm()">
                    <div class="section-a">


                      <div class="pass">
                      <input type="password" name="passw" id="ps1" onkey
down="myFunction(event)">

                      </div>

                      <div class="key backspace">
```



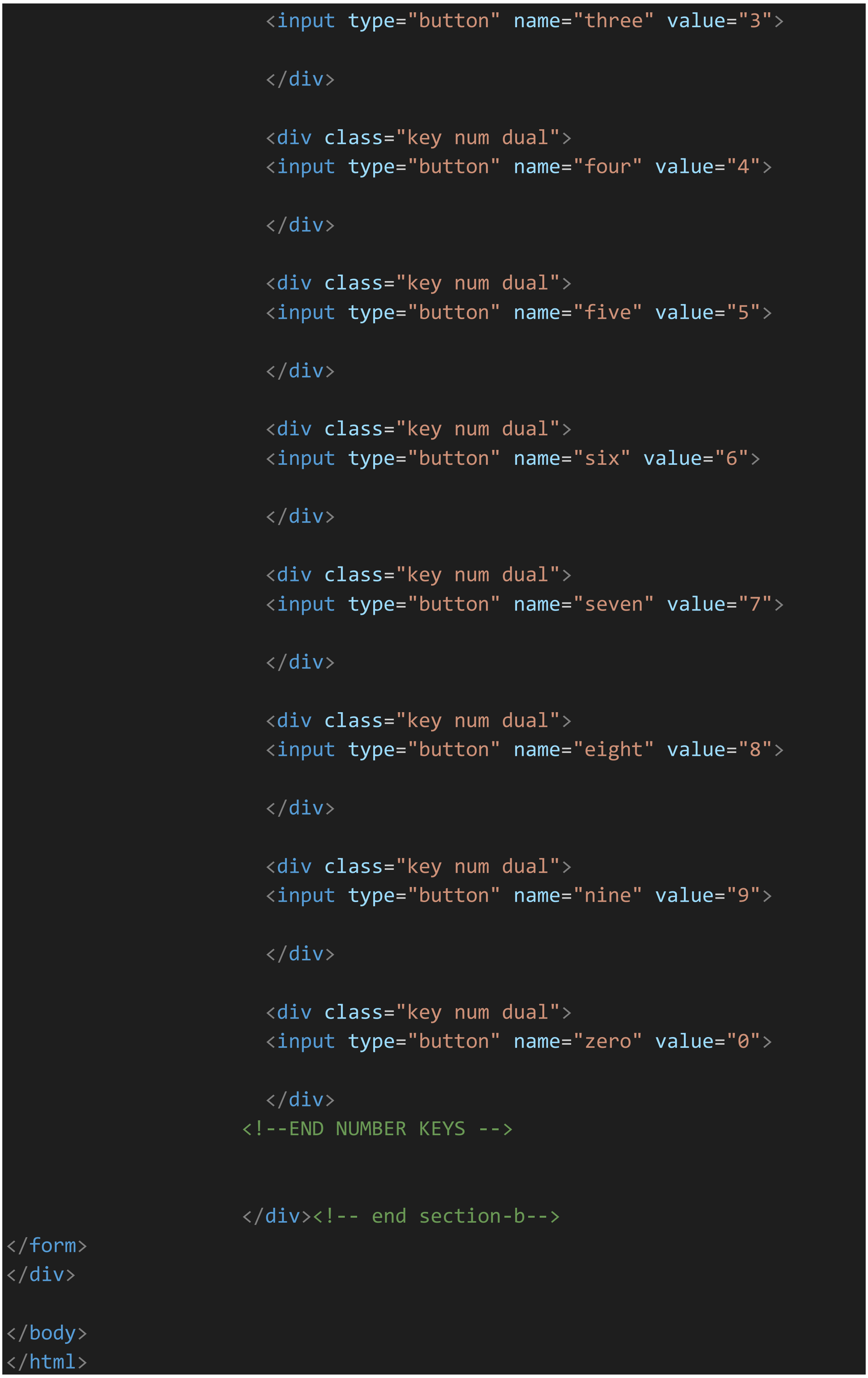
```
<input type="submit" value="SUBMIT" />
</div>

<div class="key letter">
<input type="button" name="q" value="Q">
</div>

<div class="key letter">
<input type="button" name="w" value="W">
</div>

<div class="key letter">
  <input type="button" name="e" value="E">
</div>

<div class="key letter">
  <input type="button" name="r" value="R">
</div>

<div class="key letter">
<input type="button" name="t" value="T">

</div>

<div class="key letter">
<input type="button" name="y" value="Y">

</div>

<div class="key letter">
<input type="button" name="u" value="U">

</div>

<div class="key letter">
  <input type="button" name="i" value="I">

</div>

<div class="key letter">
<input type="button" name="o" value="O">

</div>

<div class="key letter">
<input type="button" name="p" value="P">

</div>
```



```
<div class="key letter">
<input type="button" name="a" value="A">

</div>

<div class="key letter">
<input type="button" name="s" value="S">

</div>

<div class="key letter">
<input type="button" name="d" value="D">

</div>

<div class="key letter">
<input type="button" name="f" value="F">

</div>

<div class="key letter">
<input type="button" name="g" value="G">

</div>

<div class="key letter">
<input type="button" name="h" value="H">

</div>

<div class="key letter">
<input type="button" name="j" value="J">

</div>

<div class="key letter">
<input type="button" name="k" value="K">

</div>

<div class="key letter">
<input type="button" name="l" value="L">

</div>

<div class="key letter">
<input type="button" name="z" value="Z">
```



```
      </div>

      <div class="key letter">
      <input type="button" name="x" value="X">

      </div>

      <div class="key letter">
      <input type="button" name="c" value="C">

      </div>

      <div class="key letter">
      <input type="button" name="v" value="V">

      </div>

      <div class="key letter">
      <input type="button" name="b" value="B">

      </div>

      <div class="key letter">
      <input type="button" name="n" value="N">

      </div>

      <div class="key letter">
      <input type="button" name="m" value="M">

      </div>


    </div><!-- end section-a-->
    <div class="section-b">

      <div class="key num dual">
      <input type="button" name="one" value="1">

      </div>

      <div class="key num dual">
      <input type="button" name="two" value="2">

      </div>

      <div class="key num dual">
```



```
                      <input type="button" name="three" value="3">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="four" value="4">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="five" value="5">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="six" value="6">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="seven" value="7">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="eight" value="8">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="nine" value="9">

                      </div>

                      <div class="key num dual">
                      <input type="button" name="zero" value="0">

                      </div>
                    <!--END NUMBER KEYS -->


                    </div><!-- end section-b-->
</form>
</div>

</body>
</html>
```

4.3 Constraints, Alternatives and Tradeoffs

4.3.1 Alternative Approaches

i Layered Approach

The gap between the syntactic and semantic interoperability can be reduced by harnessing a layered approach that resembles the principles of networking. The layers can be categorized as- Syntax layer, object layer and the Semantic layer. Each layer has a number of sub-layers that correspond to a specific data modeling feature. The data is correctly interpreted at each layer to be later passed to the next layer. Each layer relies on a number of rules and conventions to share data with the peer systems on the network. The challenge to this approach is clearly distinguishing between the layers or features prevalent in the sub-layers. Moreover, it is difficult to establish relationship and dependencies between each layer that may exist in given systems. It is equally difficult to build flexible interfaces enabling user friendly communication between disparate systems.

ii Centralized Approach

Keeping the data at a single place or location appears to be easy to manage and control. It heads towards creation of generic model independent of any specific architecture or schema. The basic strategy used in designing such approach is to generate templates that can be dynamically changed as per the suitability of the particular application and environment. The syntactic and semantic interoperability is achieved through servers designed on highly flexible client-server architecture. Though it encompass the advantages of wide access of data, it is extensively difficult to manage and maintain the shared data across multiple platforms and applications. Moreover, centralized approach is more vulnerable with respect to data breach as it entails multiple points of access at which the user information can be submitted and received.

iii Decentralized Approach

In coordination with the heterogeneous nature of data and applications, another suitable approach considered is the decentralized approach. In this approach, each system maintains its own repository of data model and architecture independently. Various fragments of such models integrate and share the data as per their adaptability of the current environment and schemas. This approach is highly dynamic and generates unpredictable results on collaborating disparate systems. The systems communicate directly in a peer-to-peer manner at the time of sharing the data. The data can be shared without assuming shared meanings but rather enabling dynamic translations of inputted terms. This mechanism enables more functionalities and security as each system incorporates defines its own privacy policies for sharing the data. The decentralized approach gives rise to the number of policy conflicts that arise due to the existence of conflicting rules in disparate access control policies of each system.

iv Similarity-based Approach

Establishing similarities requires comparing contextual attributes or components of two or more objects, described in the given language that links the semantic and schematic level. Similarity is the confidence measure between two elements in different user hierarchies. The similarity is expressed in a mathematical number that typically range in [0-1]. A framework is proposed for ranking the users and the resources on the basis of their hierarchical positions in their organizations. The relevant and authorized access is matched on merging of disparate access privileges of the users and permissions granted accordingly. It evaluates the degree of similarity between two or more user’s ranks from defined attributes in the access control policies of each user. The similarity score obtained on measuring the hierarchical distance between each user generates a unique value utilized as security level (SL) for each user/resource attribute in the defined access control policies of heterogeneous healthcare systems. A similarity score 0 implies that no matching holds between the attributes. The framework extends the usage of similarity score by defining the authorization on the basis of the hierarchical positions and the roles defined in the given policies.

4.3.2 Tradeoffs and Disadvantages of the Security approach

For encryption, this cipher needs a key with the same length as the length of the original data. For example, to encrypt a hard disk, we require a second hard disk of the same size to store the key. Another disadvantage of this cipher is that the data of the key has to be, ideally, completely randomly chosen. Most computers we use today are not capable of generating such truly random keys.

5. SCHEDULE, TASKS AND MILESTONES

5.1 REVIEW 1

5.1.1 Outcomes:

Dataset Preparation

Data Pre – Processing

5.1.2 Time elapsed: 1 month

5.2 REVIEW 2

5.2.1 Outcomes:

Data Integration

Data Analytics

Data Modelling

Data Visualization

5.2.2 Time elapsed: 3 months

5.2 FINAL REVIEW

5.2.1 Outcomes:

Data Visualization

Security Module

Documentation

5.2.2 Time elapsed: 5 months

6. PROJECT DEMONSTRATION

6.1 Data Insights

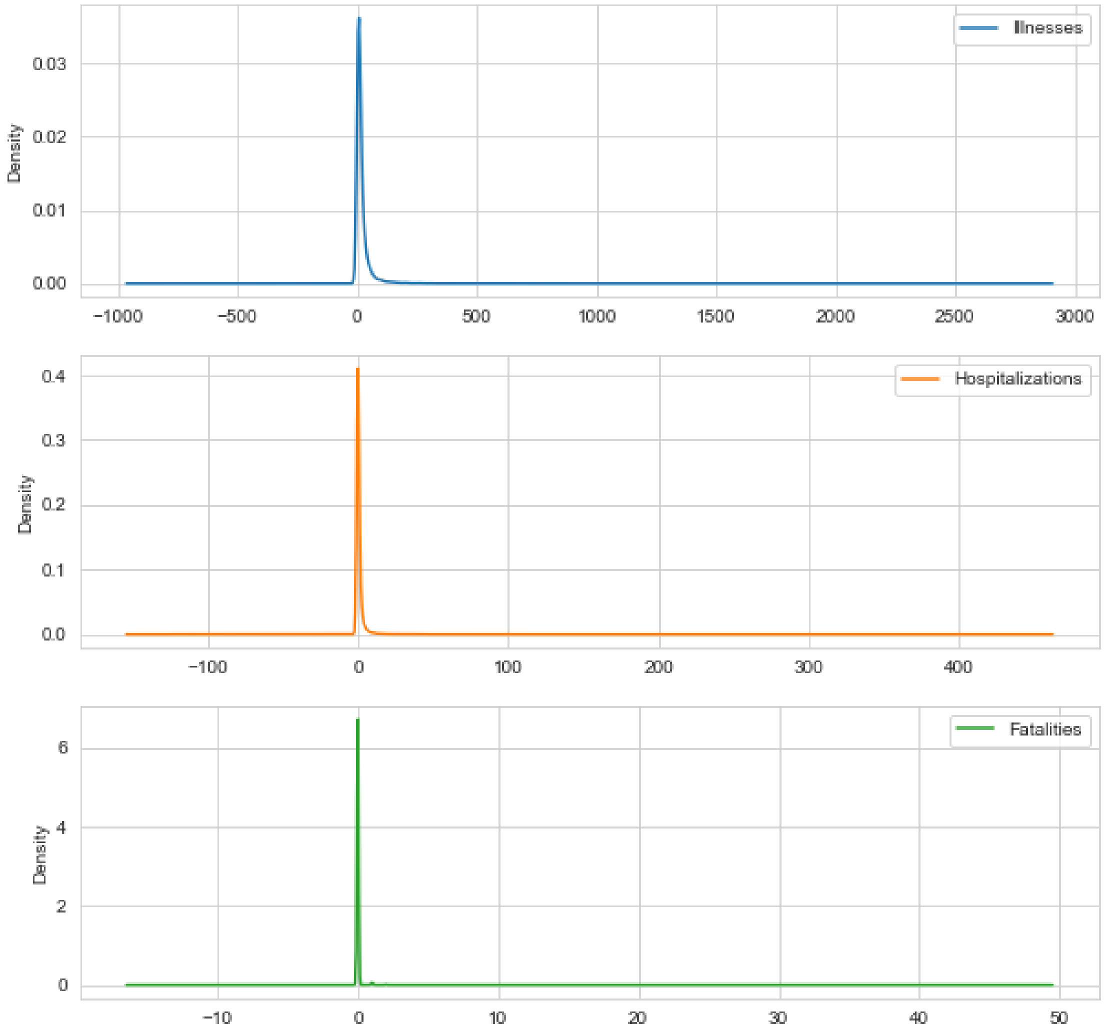


Fig 6.1.1: Density plots for major parameters

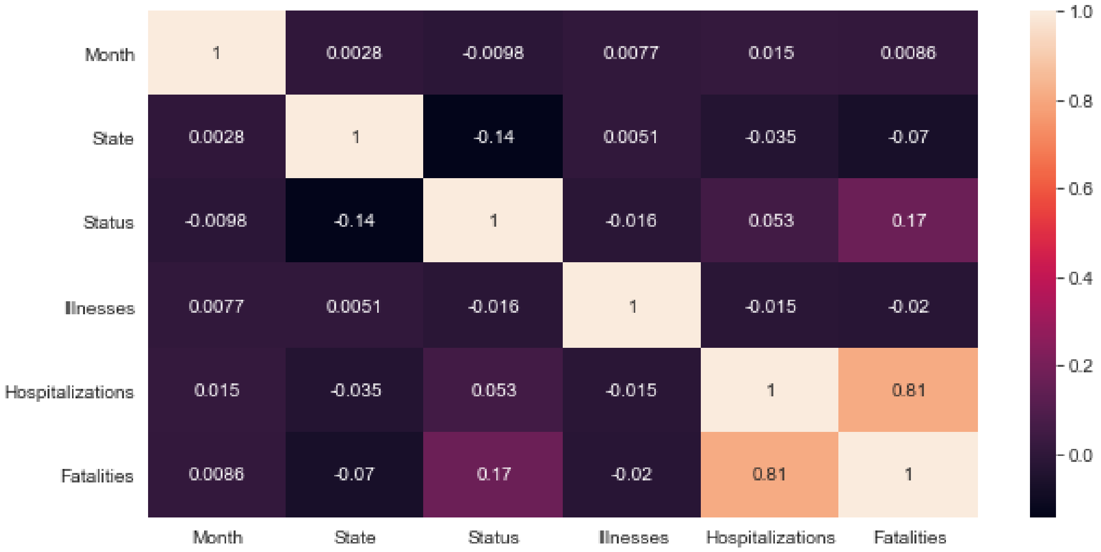


Fig 6.1.2: Heat Map for visualizing correlation

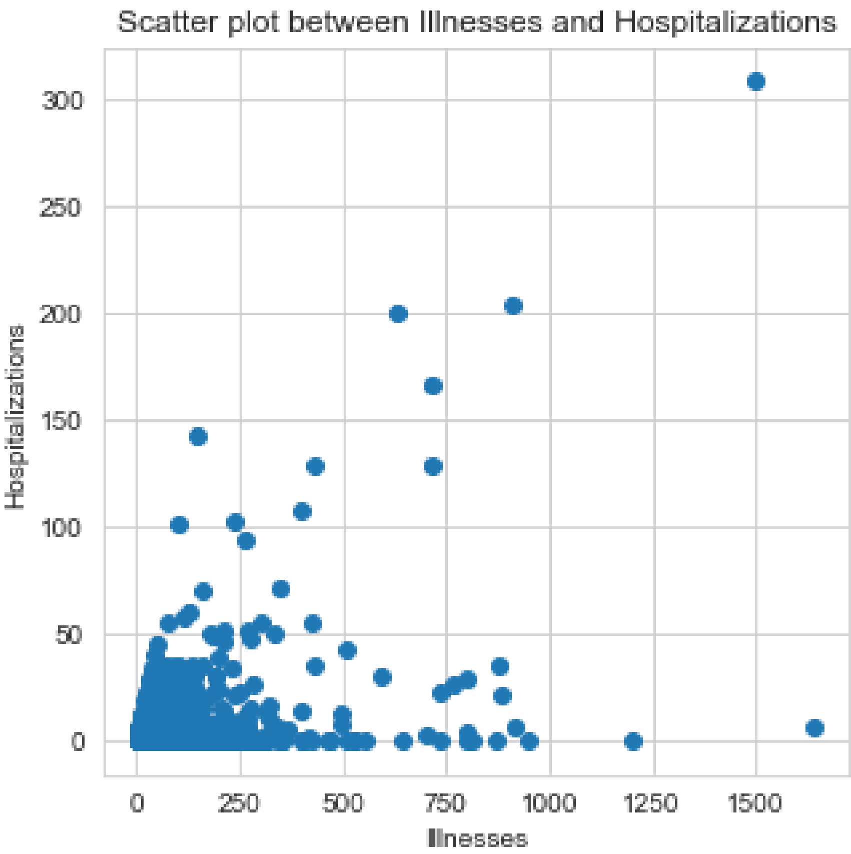


Fig 6.1.3: Scatter plot between Illnesses and Hospitalizations

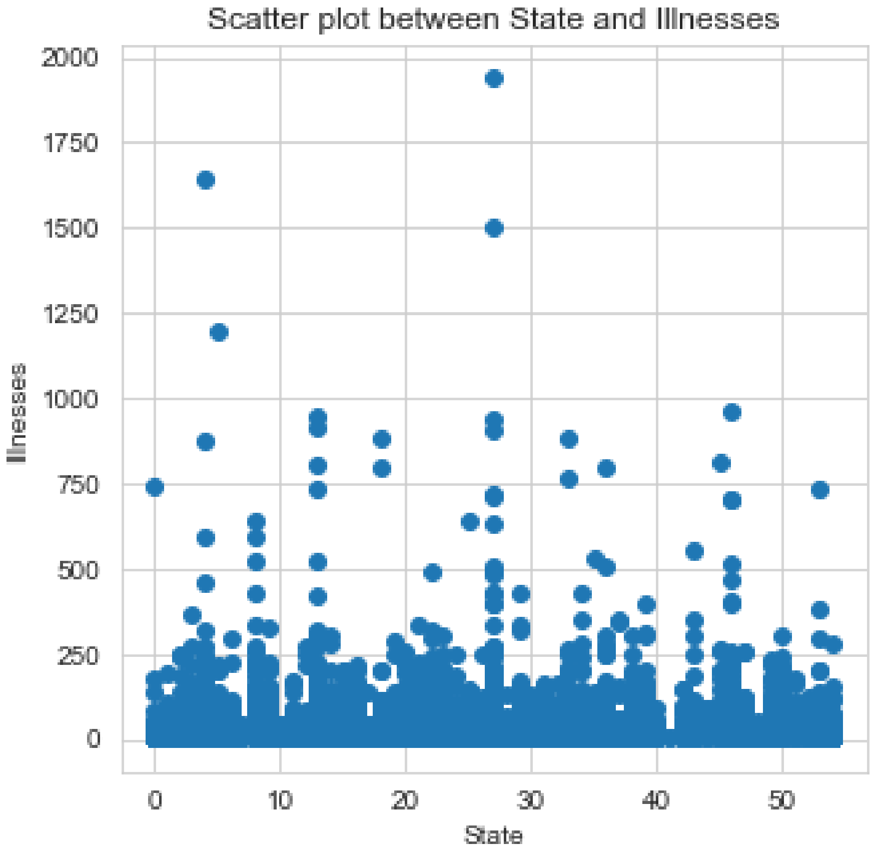


Fig 6.1.4: Scatter plot between State and Illnesses

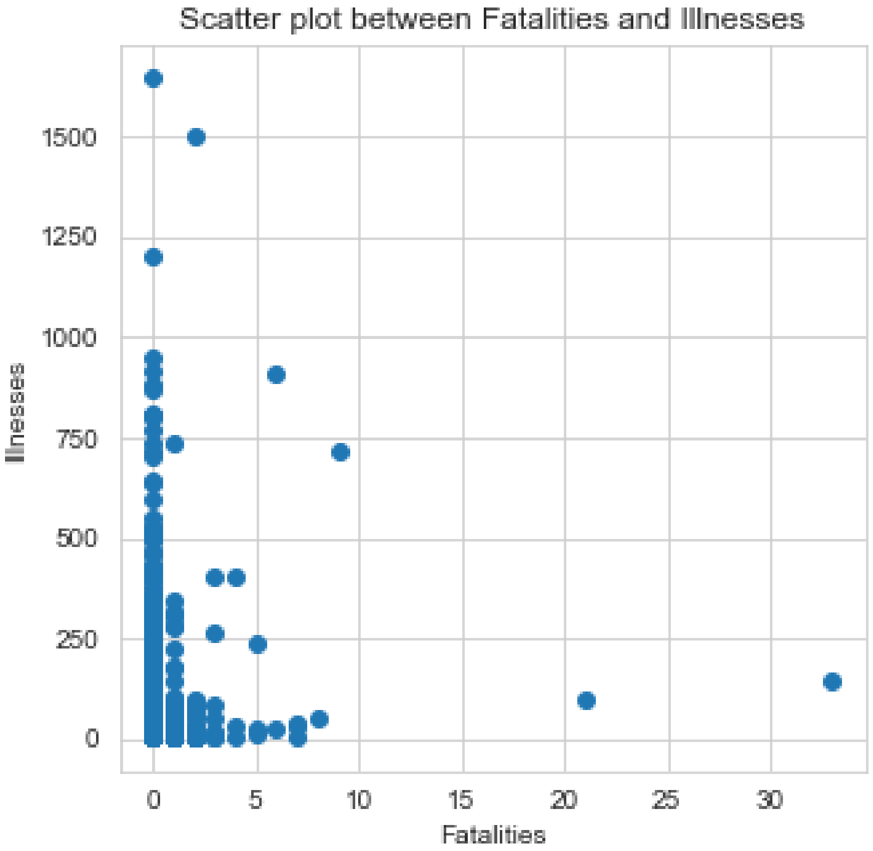


Fig 6.1.5: Scatter plot between Fatalities and Illnesses

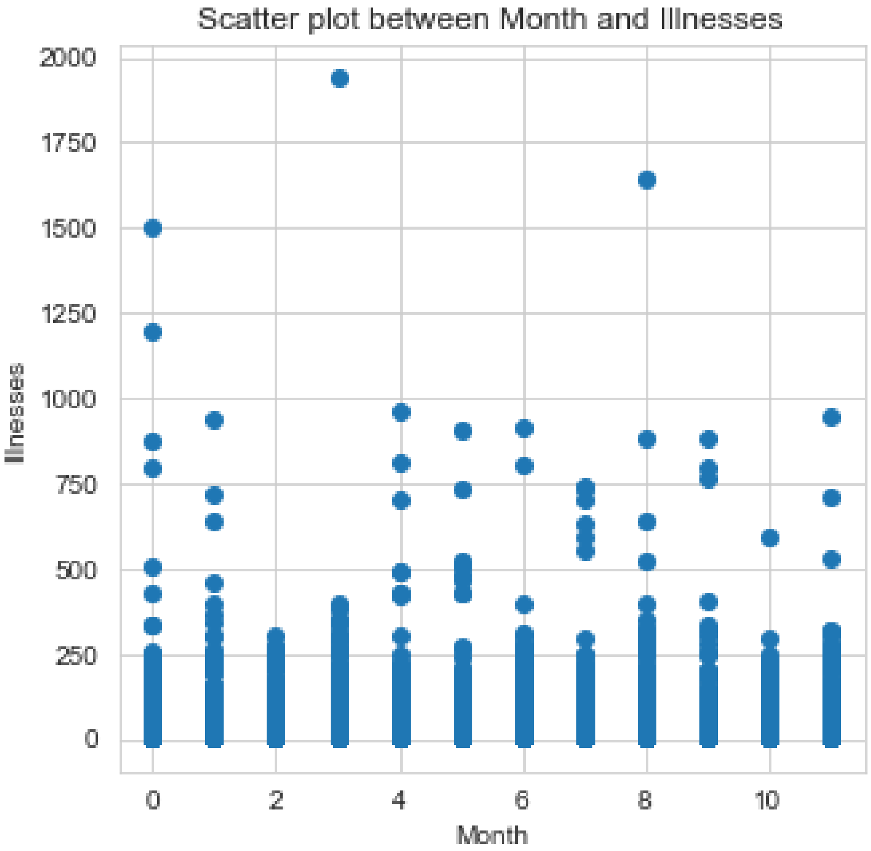


Fig 6.1.6: Scatter plot between Month and Illnesses

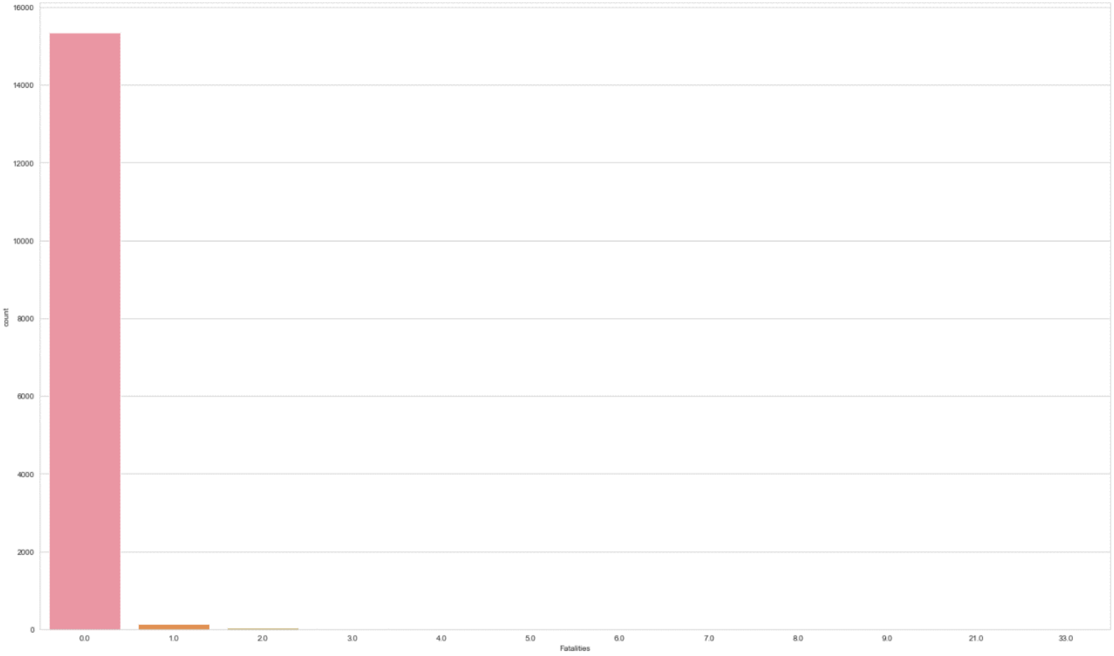


Fig 6.1.7: Bar graph showing food bourne diseases are not deadly as minimal fatalities are visible

6.2 Classification and Prediction

6.2.1 Logistic Regression

```
Training results:

Accuracy Score: 0.7904

Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      1.00      0.88     10579
         1.0       0.00      0.00      0.00      1290
         2.0       0.00      0.00      0.00       582
         3.0       0.00      0.00      0.00       291
         4.0       0.00      0.00      0.00       164
         5.0       0.00      0.00      0.00       115
         6.0       0.00      0.00      0.00        79
         7.0       0.00      0.00      0.00        49
         8.0       0.00      0.00      0.00        42
         9.0       0.00      0.00      0.00        24
        10.0       0.00      0.00      0.00        20


    accuracy                           0.79     13383
   macro avg       0.01      0.02      0.02     13383
weighted avg       0.62      0.79      0.70     13383


Confusion Matrix:
[[10578     0     1 ...     0     0     0]
 [ 1290     0     0 ...     0     0     0]
 [  582     0     0 ...     0     0     0]]


Average Accuracy:     0.7904

Standard Deviation:   0.0004


Test results:

Accuracy Score: 0.7821

Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      1.00      0.88      4485
         1.0       1.00      0.00      0.00       596
         2.0       0.00      0.00      0.00       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16
    accuracy                           0.78      5736
   macro avg       0.05      0.03      0.02      5736
weighted avg       0.72      0.78      0.69      5736


Confusion Matrix:
[[4485    0    0 ...    0    0    0]
 [ 595    1    0 ...    0    0    0]
 [ 262    0    0 ...    0    0    0]]
```

6.2.2 SVC

```
Training results:

Accuracy Score: 0.7906

Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      1.00      0.88     10579
         1.0       0.00      0.00      0.00      1290
         2.0       0.00      0.00      0.00       582
         3.0       0.00      0.00      0.00       291
         4.0       0.00      0.00      0.00       164
         5.0       0.00      0.00      0.00       115
         6.0       0.00      0.00      0.00        79
         7.0       1.00      0.02      0.04        49
         8.0       0.00      0.00      0.00        42
         9.0       0.00      0.00      0.00        24
        10.0       0.00      0.00      0.00        20

    accuracy                           0.79     13383
   macro avg       0.05      0.04      0.03     13383
weighted avg       0.63      0.79      0.70     13383


Confusion Matrix:
[[10579     0     0 ...     0     0     0]
 [ 1290     0     0 ...     0     0     0]
 [  582     0     0 ...     0     0     0]]

Average Accuracy:      0.7905

Standard Deviation:    0.0003

Test results:
```

```
Accuracy Score: 0.7819

Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      1.00      0.88      4485
         1.0       0.00      0.00      0.00       596
         2.0       0.00      0.00      0.00       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16


    accuracy                           0.78      5736
   macro avg       0.02      0.03      0.02      5736
weighted avg       0.61      0.78      0.69      5736


Confusion Matrix:
[[4485    0    0 ...    0    0    0]
 [ 596    0    0 ...    0    0    0]
 [ 262    0    0 ...    0    0    0]]
```

6.2.3 K-Nearest Neighbors

```
Training results:

Accuracy Score: 0.7864

Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      0.99      0.88     10579
         1.0       0.19      0.02      0.03      1290
         2.0       0.19      0.01      0.02       582
         3.0       0.50      0.01      0.01       291
         4.0       0.00      0.00      0.00       164
         5.0       0.40      0.02      0.03       115
         6.0       0.50      0.01      0.02        79
         7.0       0.00      0.00      0.00        49
         8.0       0.00      0.00      0.00        42
         9.0       0.00      0.00      0.00        24
```

```
        10.0       0.00      0.00      0.00        20

    accuracy                           0.79     13383
   macro avg       0.05      0.02      0.02     13383
weighted avg       0.67      0.79      0.70     13383


Confusion Matrix:
[[10492    76     9 ...     0     0     0]
 [ 1260    23     6 ...     0     0     0]
 [  572     5     5 ...     0     0     0]]


Average Accuracy:      0.7835

Standard Deviation:    0.0113


Test results:

Accuracy Score: 0.7718

Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      0.99      0.87      4485
         1.0       0.07      0.01      0.02       596
         2.0       0.08      0.00      0.01       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16

    accuracy                           0.77      5736
   macro avg       0.02      0.03      0.02      5736
weighted avg       0.62      0.77      0.68      5736


Confusion Matrix:
[[4421   56    4 ...    0    0    0]
 [ 587    5    3 ...    0    0    0]
 [ 257    3    1 ...    0    0    0]]
```

6.2.4 Naïve Bayes Classifier

```
Training results:

Accuracy Score: 0.7750
```

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      0.98      0.88     10579
         1.0       0.00      0.00      0.00      1290
         2.0       0.00      0.00      0.00       582
         3.0       0.00      0.00      0.00       291
         4.0       0.00      0.00      0.00       164
         5.0       0.00      0.00      0.00       115
         6.0       0.00      0.00      0.00        79
         7.0       0.02      0.04      0.02        49
         8.0       0.00      0.00      0.00        42
         9.0       0.00      0.00      0.00        24
        10.0       0.00      0.00      0.00        20

    accuracy                           0.78     13383
   macro avg       0.28      0.43      0.31     13383
weighted avg       0.63      0.78      0.69     13383


Confusion Matrix:
[[10346     0     0 ...     0     0     0]
 [ 1262     0     0 ...     0     0     0]
 [  572     0     0 ...     0     0     0]]


Average Accuracy:     0.7740

Standard Deviation:   0.0049

Test results:

Accuracy Score: 0.7667

Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      0.98      0.87      4485
         1.0       0.00      0.00      0.00       596
         2.0       0.00      0.00      0.00       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16

    accuracy                           0.77      5736
   macro avg       0.02      0.02      0.02      5736
weighted avg       0.61      0.77      0.68      5736


Confusion Matrix:
[[4398    0    0 ...    1    0    0]
 [ 579    0    0 ...    0    1    0]
 [ 253    0    0 ...    0    0    0]]
```

6.2.5 Decision Tree Classifier

```
Training results:

Accuracy Score: 0.7949

Classification Report:
              precision    recall  f1-score   support

         0.0       0.79      1.00      0.89     10579
         1.0       0.67      0.01      0.02      1290
         2.0       0.64      0.01      0.02       582
         3.0       0.60      0.01      0.02       291
         4.0       1.00      0.01      0.01       164
         5.0       0.40      0.02      0.03       115
         6.0       0.50      0.01      0.02        79
         7.0       1.00      0.08      0.15        49
         8.0       0.00      0.00      0.00        42
         9.0       1.00      0.04      0.08        24
        10.0       1.00      0.05      0.10        20

    accuracy                           0.79     13383
   macro avg       0.60      0.33      0.36     13383
weighted avg       0.77      0.79      0.71     13383


Confusion Matrix:
[[10574     2     1 ...     0     0     0]
 [ 1278    12     0 ...     0     0     0]
 [  573     2     7 ...     0     0     0]
 ...
 [    0     0     0 ...     1     0     0]
 [    0     0     0 ...     0     1     0]
 [    0     0     0 ...     0     0     1]]

Average Accuracy:     0.7873

Standard Deviation:   0.0025

Test results:

Accuracy Score: 0.7791

Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      1.00      0.88      4485
         1.0       0.10      0.00      0.00       596
         2.0       0.00      0.00      0.00       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16

    accuracy                           0.78      5736
   macro avg       0.02      0.02      0.02      5736
weighted avg       0.62      0.78      0.69      5736


Confusion Matrix:
[[4468    8    1 ...    0    0    0]
 [ 593    1    0 ...    0    0    0]
 [ 260    0    0 ...    0    0    0]]
```

6.2.6 Random Forest Classifier

```
Training results:

Accuracy Score: 0.7948

Classification Report:
              precision    recall  f1-score   support

         0.0       0.80      1.00      0.89     10579
         1.0       0.58      0.01      0.02      1290
         2.0       0.75      0.01      0.02       582
         3.0       0.50      0.02      0.03       291
         4.0       0.67      0.01      0.02       164
         5.0       0.38      0.03      0.05       115
         6.0       0.00      0.00      0.00        79
         7.0       1.00      0.08      0.15        49
         8.0       0.00      0.00      0.00        42
         9.0       1.00      0.04      0.08        24
        10.0       0.00      0.00      0.00        20

    accuracy                           0.79     13383
   macro avg       0.60      0.36      0.38     13383
weighted avg       0.75      0.79      0.71     13383


Confusion Matrix:
[[10568     4     1 ...     0     0     0]
 [ 1275    14     0 ...     0     0     0]
 [  573     2     6 ...     0     0     0]]

Average Accuracy:      0.7852

Standard Deviation:    0.0023

Test results:

Accuracy Score: 0.7779
```

```
Classification Report:
              precision    recall  f1-score   support

         0.0       0.78      0.99      0.88      4485
         1.0       0.10      0.00      0.01       596
         2.0       0.00      0.00      0.00       262
         3.0       0.00      0.00      0.00       131
         4.0       0.00      0.00      0.00        65
         5.0       0.00      0.00      0.00        44
         6.0       0.00      0.00      0.00        27
         7.0       0.00      0.00      0.00        26
         8.0       0.00      0.00      0.00        11
         9.0       0.00      0.00      0.00        11
        10.0       0.00      0.00      0.00        16

    accuracy                           0.78      5736
   macro avg       0.02      0.02      0.02      5736
weighted avg       0.62      0.78      0.69      5736


Confusion Matrix:
[[4460   15    1 ...    0    0    0]
 [ 592    2    0 ...    0    0    0]
 [ 260    0    0 ...    0    0    0]]
```

6.3 Training the ML Model

```python
from math import floor
def partition(vector, fold, k):

    size = vector.shape[0]

    start = floor((size/k)*fold)

    end = floor((size/k)*(fold+1))

    validation = vector[start:end]

    #print(str(type(vector)))

    if str(type(vector)) == "<class 'scipy.sparse.csr.csr_matrix'>":

        indices = range(start, end)

        mask = numpy.ones(vector.shape[0], dtype=bool)

        mask[indices] = False

        training = vector[mask]
    elif str(type(vector)) == "<class 'numpy.ndarray'>":
        training = numpy.concatenate((vector[:start], vector[end:]))
    return training, validation


def Cross_Validation(learner, k, examples, labels):
    train_folds_score = []
    validation_folds_score = []
    test_score_auc = []
    test_score_mcc = []
    for fold in range(0, k):
        training_set, validation_set = partition(examples, fold, k)
        training_labels, validation_labels = partition(labels, fold,
 k)
        learner.fit(training_set, training_labels)
        training_predicted = learner.predict(training_set)
        validation_predicted = learner.predict(validation_set)
        test_predicted = learner.predict(X_test)
        train_folds_score.append(roc_auc_score(training_labels, trai
ning_predicted))
        validation_folds_score.append(roc_auc_score(validation_label
s, validation_predicted))
        test_score_auc.append(roc_auc_score(Y_test, test_predicted))
        test_score_mcc.append(matthews_corrcoef(Y_test, test_predict
ed))
    return train_folds_score, validation_folds_score, test_score_auc
, test_score_mcc
```

In [23]:

```
def run(model) :
    train_scores, validation_scores, test_scores_auc, test_scores_mc
c = Cross_Validation(model, 10, X_test, Y_test)
    #print(train_scores, validation_scores, test_scores)
    print(model)
    print('Train AUC', float(format(numpy.mean(train_scores), '.3f')
))
    print('Validation AUC',float(format(numpy.mean(validation_scores
), '.3f')))
    print('Test AUC',float(format(numpy.mean(test_scores_auc), '.3f'
)))
    print('Test MCC',float(format(numpy.mean(test_scores_mcc), '.3f'
)))
    print()
```

In [25]:

```
model = Sequential()
model.add(Dense(256, input_dim=1, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

In [ ]:

```
history = model.fit(X_train, Y_train, epochs=100, batch_size=4, verb
ose=1, validation_split=0.1)
```

```
Train on 15486 samples, validate on 1721 samples
Epoch 1/100
15486/15486 [==============================] - 55912s 4s/step - loss:
 0.1537 - accuracy: 0.1620 - val_loss: 0.2512 - val_accuracy: 0.0924
Epoch 2/100
15486/15486 [==============================] - 20376s 1s/step - loss:
 -0.9542 - accuracy: 0.2514 - val_loss: -0.9563 - val_accuracy: 0.579
3
Epoch 3/100
15486/15486 [==============================] - 22259s 1s/step - loss:
 -2.5810 - accuracy: 0.3733 - val_loss: -0.8726 - val_accuracy: 0.092
4
Epoch 4/100
15486/15486 [==============================] - 19561s 1s/step - loss:
 -3.0549 - accuracy: 0.4407 - val_loss: -1.1454 - val_accuracy: 0.092
4
Epoch 5/100
```

```
15486/15486 [==============================] - 24930s 2s/step - loss:
 -3.2108 - accuracy: 0.4725 - val_loss: -1.4446 - val_accuracy: 0.567
7
Epoch 6/100
15486/15486 [==============================] - 20879s 1s/step - loss:
 -3.3477 - accuracy: 0.5033 - val_loss: -1.3890 - val_accuracy: 0.616
5
Epoch 7/100
15486/15486 [==============================] - 20974s 1s/step - loss:
 -3.3401 - accuracy: 0.5138 - val_loss: -1.3531 - val_accuracy: 0.645
0
Epoch 8/100
15486/15486 [==============================] - 19218s 1s/step - loss:
 -3.3214 - accuracy: 0.5265 - val_loss: -1.4889 - val_accuracy: 0.549
1
Epoch 9/100
15486/15486 [==============================] - 14305s 924ms/step - lo
ss: -3.2661 - accuracy: 0.5303 - val_loss: -1.4542 - val_accuracy: 0.
6049
Epoch 10/100
15486/15486 [==============================] - 21324s 1s/step - loss:
 -3.4066 - accuracy: 0.5426 - val_loss: -1.3945 - val_accuracy: 0.637
4
Epoch 11/100
15486/15486 [==============================] - 23875s 2s/step - loss:
 -3.4112 - accuracy: 0.5447 - val_loss: -1.4849 - val_accuracy: 0.579
3
```

6.4 Analyzing Security while entering EHR

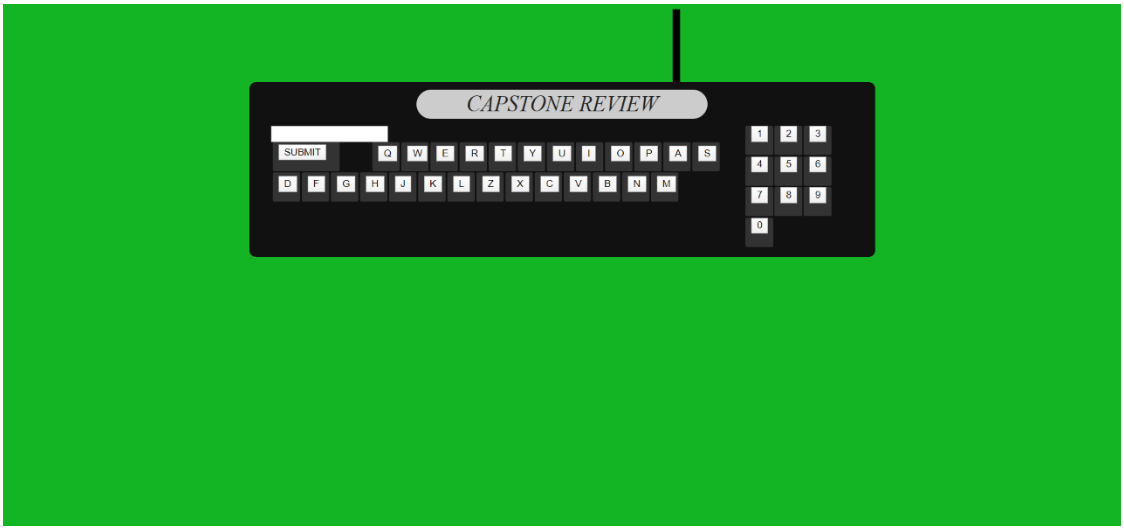
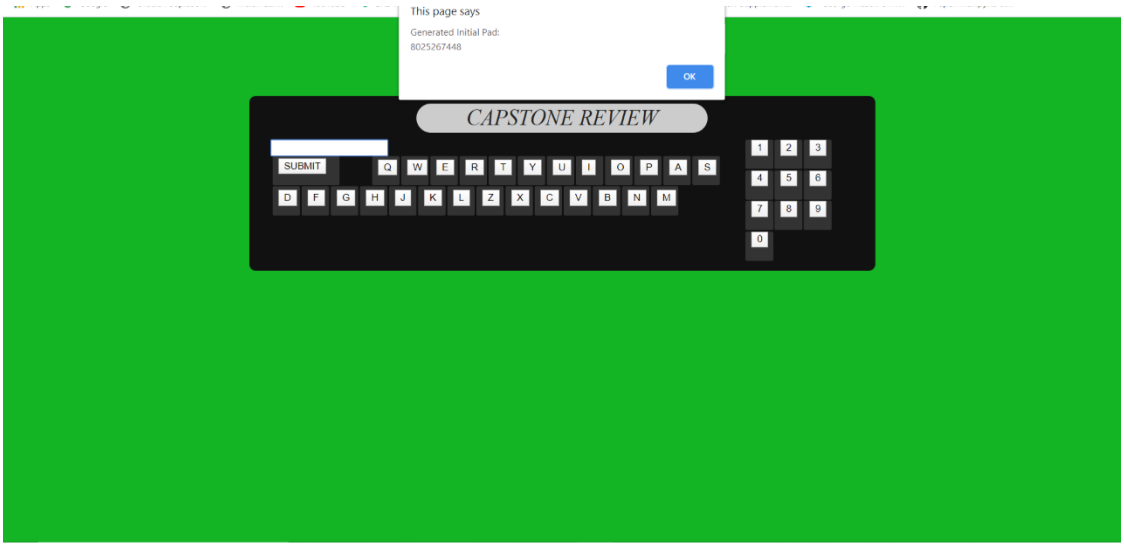


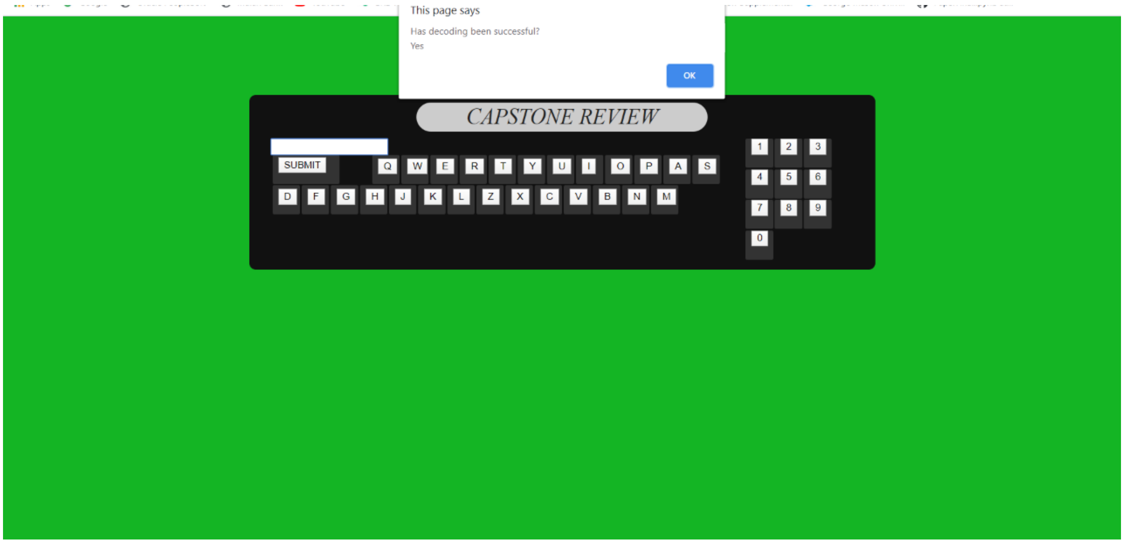
Fig 6.4.1: EHR Interface with Vernam Cipher



This page says

Generated Initial Pad:
8025267448

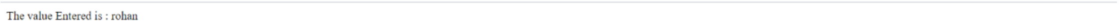
Fig 6.4.2: EHR Interface gives a message after Generating Initial Pad



This page says

Has decoding been successful?
Yes

Fig 6.4.3: EHR Interface gives a message after successfully decoding using the one time pad



The value Entered is : rohan

Fig 6.4.4: EHR Interface displays the data after submit button is pressed

7. RESULT & DISCUSSION

7.1 Logistic Regression

7.1.1 Training

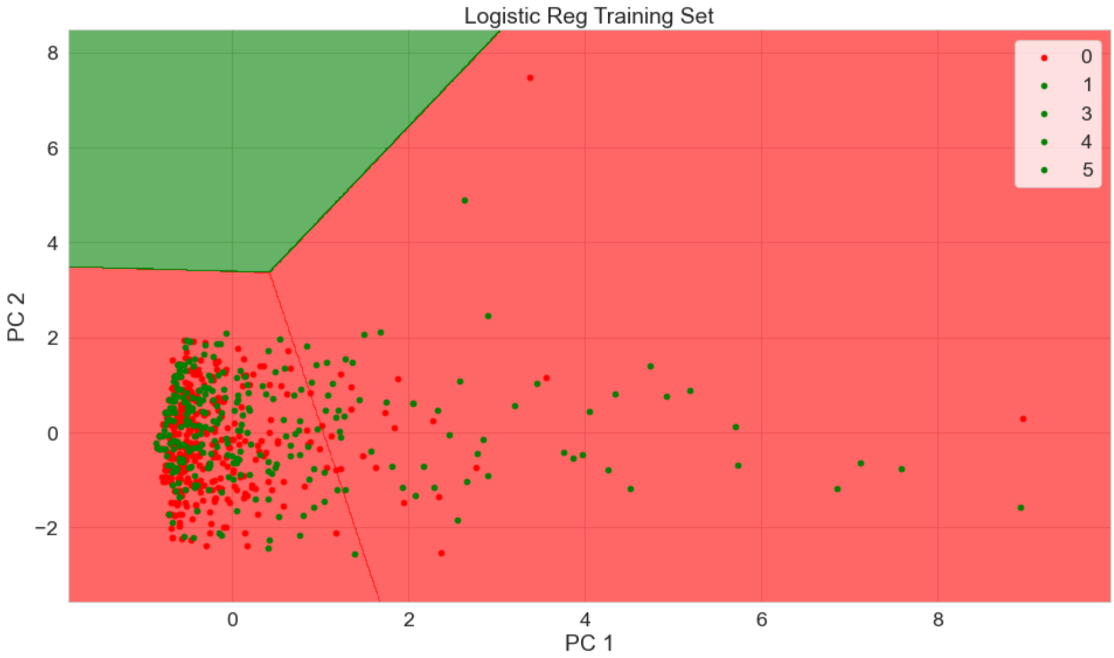


Fig 7.1.1: Coloured Scatter plot for training using Logistic Regression Classifier

7.1.2 Testing

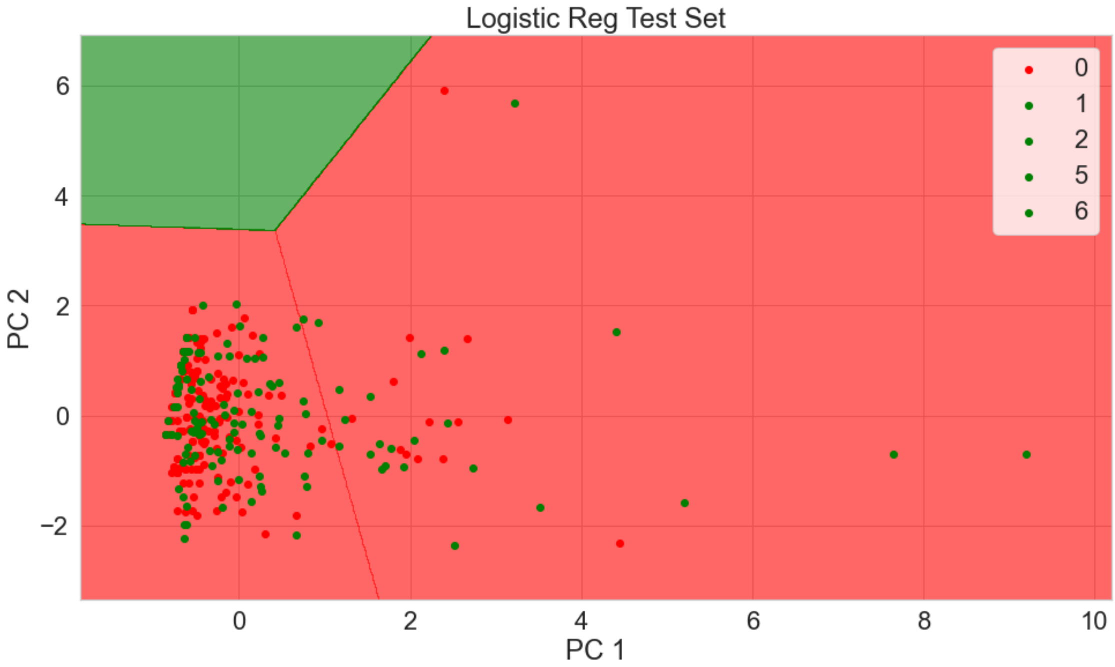


Fig 7.1.2: Coloured Scatter plot for testing using Logistic Regression Classifier

7.2 SVC

7.2.1 Training

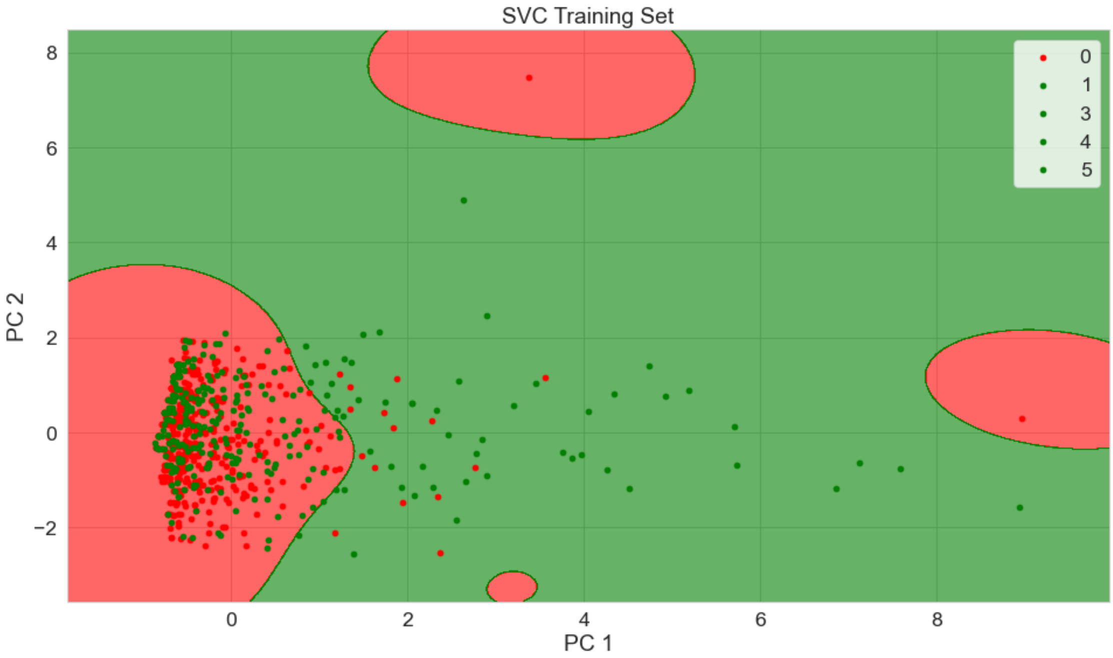


Fig 7.2.1: Coloured Scatter plot for training using Support Vector Classifier

7.2.2 Testing

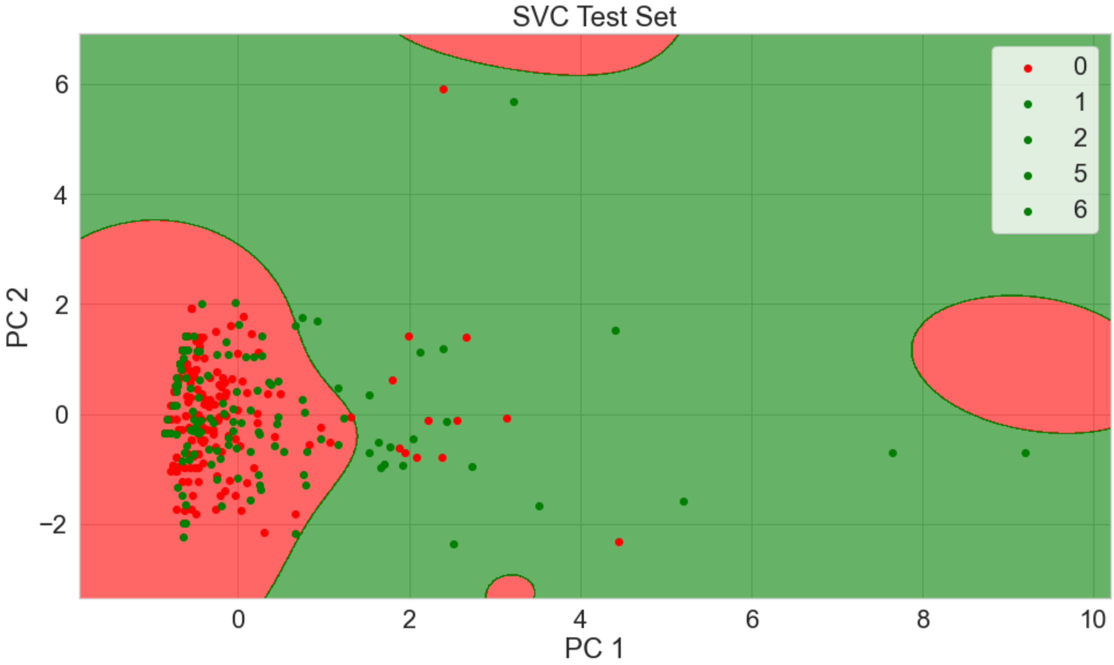


Fig 7.2.2: Coloured Scatter plot for testing using Support Vector Classifier

7.3 K – Nearest Neighbour

7.3.1 Training

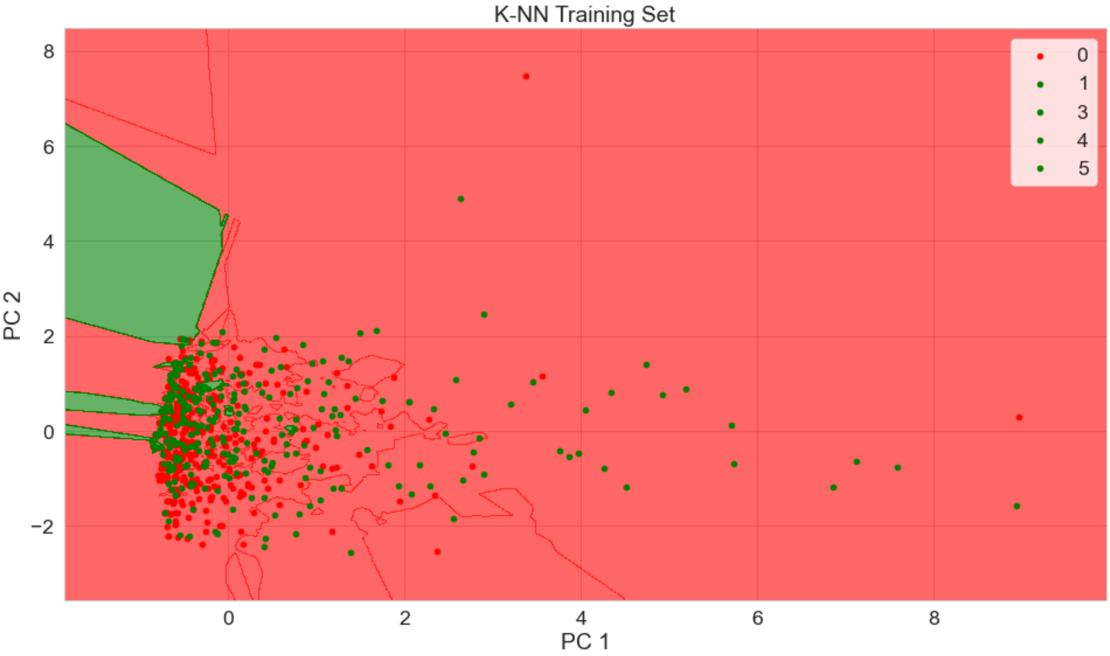


Fig 7.3.1: Coloured Scatter plot for training using K-NN Classifier

7.3.2 Testing

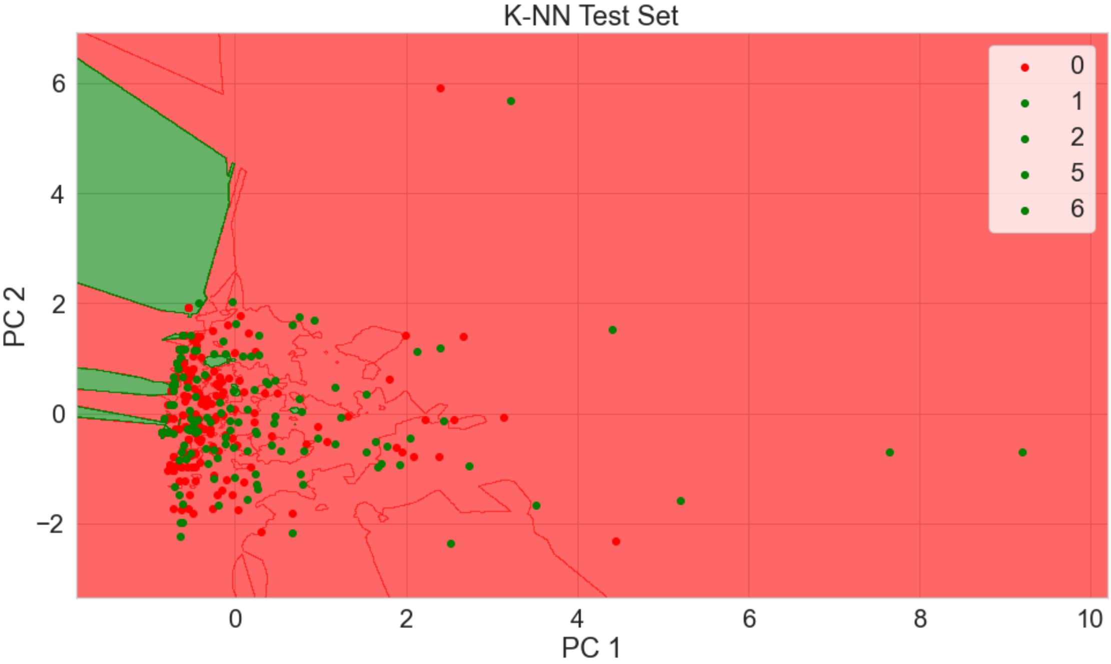


Fig 7.3.2: Coloured Scatter plot for testing using K-NN Classifier

7.4 Naïve Bayes

7.4.1 Training

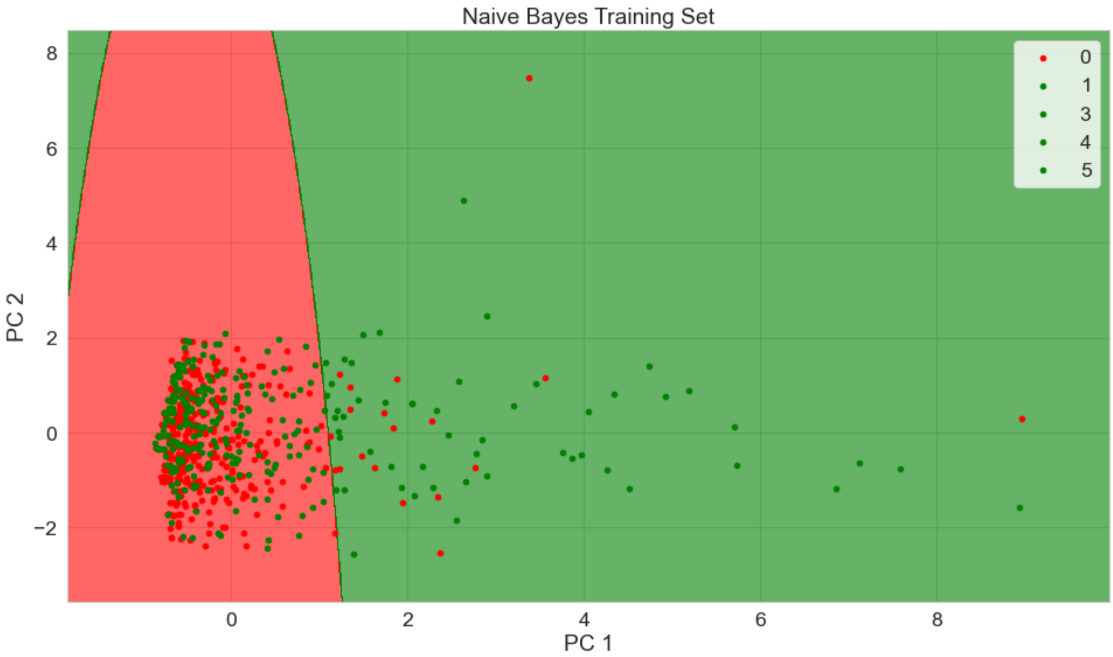


Fig 7.4.1: Coloured Scatter plot for training using Naïve Bayes Classifier

7.4.2 Testing

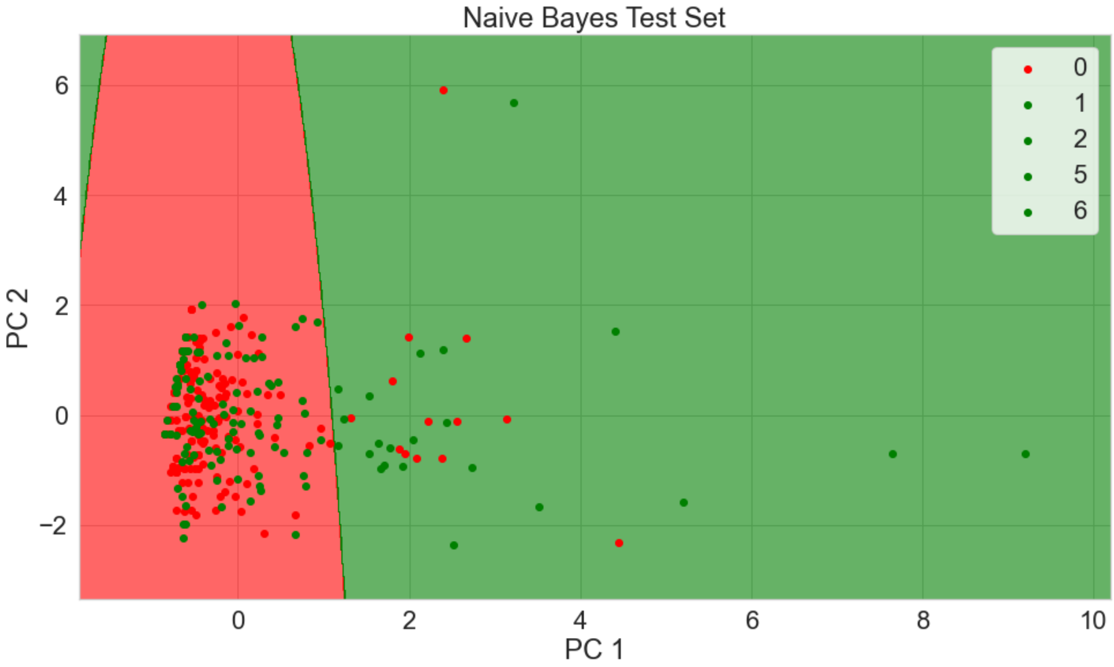


Fig 7.4.2: Coloured Scatter plot for testing using Naïve Bayes Classifier

7.5 Decision Tree

7.5.1 Training

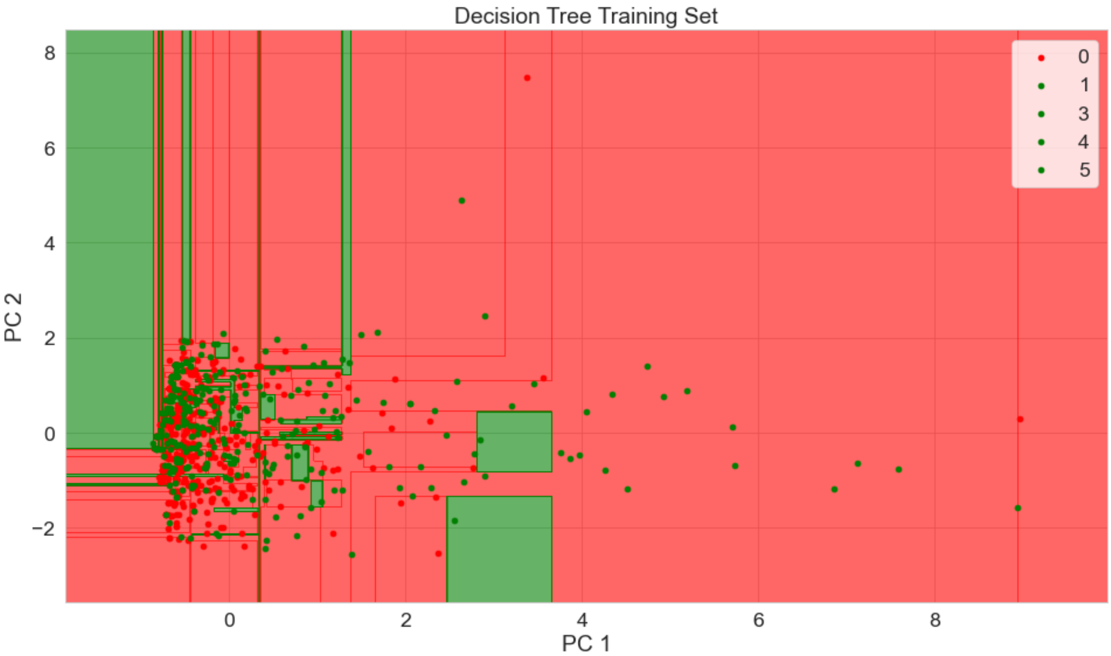


Fig 7.5.1: Coloured Scatter plot for training using Decision Tree Classifier

7.5.2 Testing

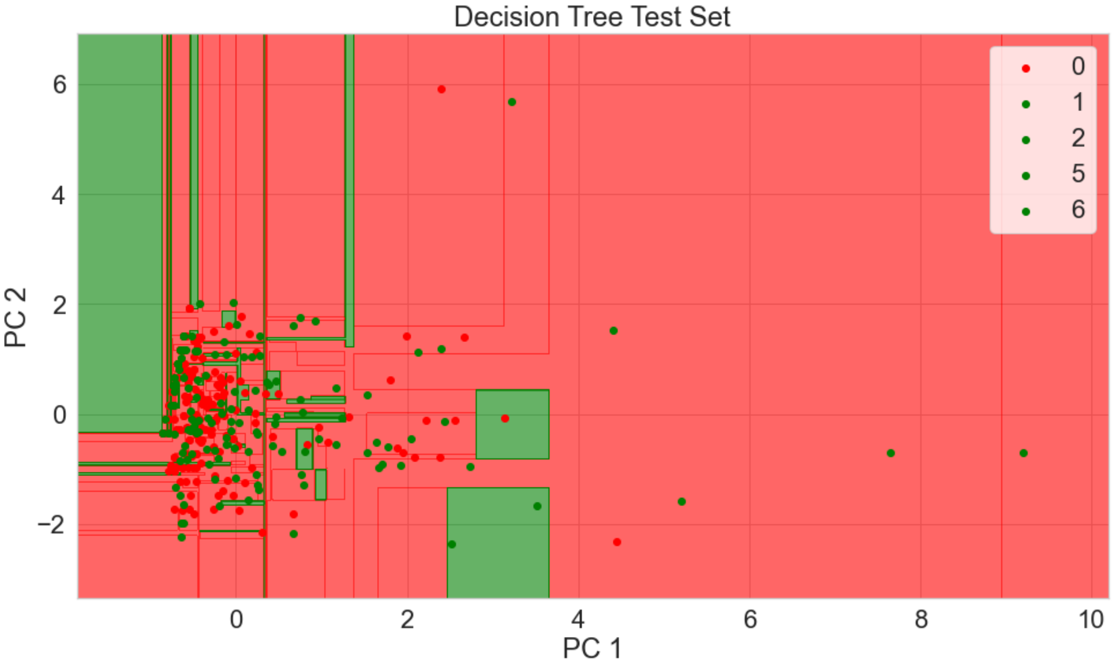


Fig 7.5.2: Coloured Scatter plot for training using Decision Tree Classifier

7.6 Random Forest

7.6.1 Training

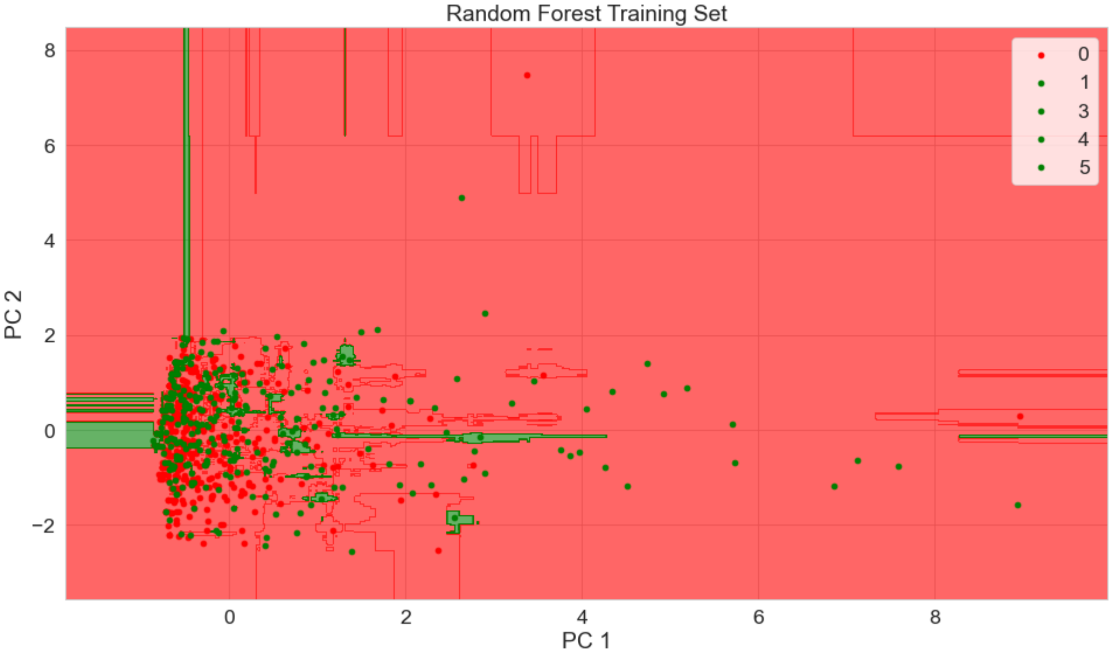


Fig 7.6.1: Coloured Scatter plot for training using Random Forest Classifier

7.6.2 Testing

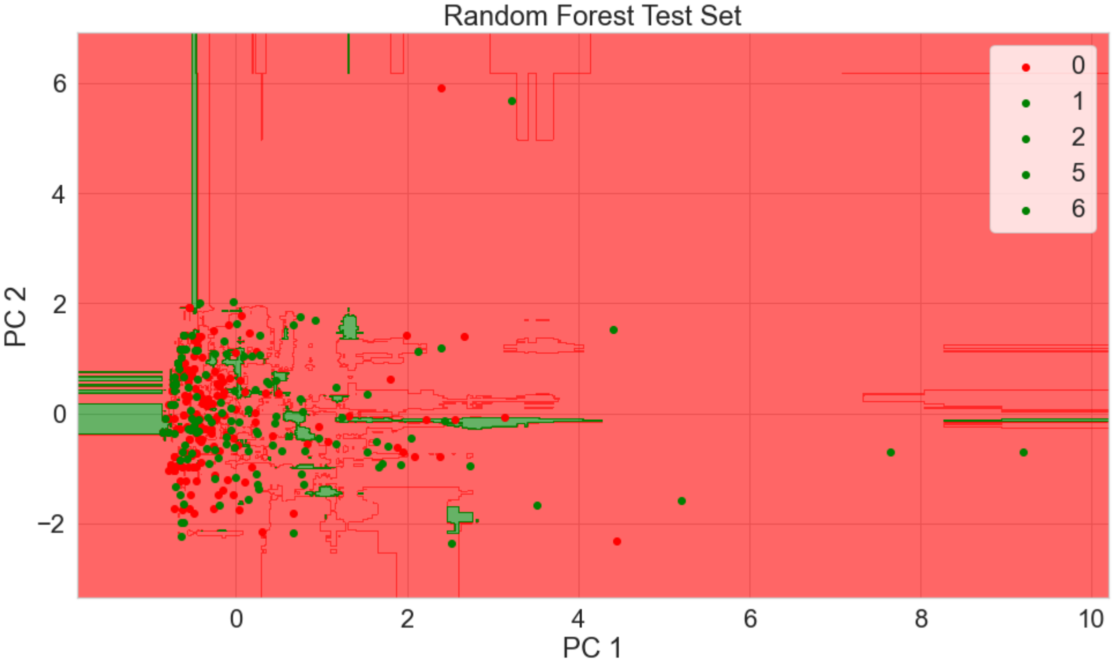


Fig 7.6.2: Coloured Scatter plot for training using Random Forest Classifier

I intended to implement numerous classification algorithms to build models for prediction and demonstrate a cryptographic algorithm, which could lead to safeguarding sensitive patient information now more than ever when medical data is going to be researched a lot more than ever.
Six classification models were tested for prediction accuracy and sensitivity and it was demonstrated that the Random Forest Algorithm gives the highest accuracy and sensitivity (a score of 79.49) among the predictive algorithms. As a future work, I will extend this study to include future engineering methods, to measure if the predictive power of the models could be increased or not. Also, since the principal component chosen is State of the Patient and the project aims to simplify EHR documentation, this predictive model could be extrapolated to aid in COVID19 research in the upcoming years.

8. SUMMARY

In this study, I aimed to learn the accuracy with which I could predict whether the Food Borne Diseases stated in the data sets causes significant illness, at the same time enhance the security aspect while dealing with EHR. In this paper I proposed a predictive sequential model with considerable functional accuracy along with a system which uses a different secret keyboard mechanism with an aim to prevent Shoulder Surfing attacks. By using this method of authentication, users can easily log into their systems without worrying about the Shoulder Surfers as the character which is displayed on the screen is not the same as the character which is being stored in the back end. It is a simple and effective solution even for camera based attacks. In future, this system can be used for applications which require moderate to high security, since the other conventional password schemes are vulnerable to shoulder surfing. Overall, my main goal for this research is to come up with a solution which uses existing cipher to solve modern problems.

References

[1] IEEE, Standards glossary, 2013. In IEEE Retrieved from http://www.ieee.org/education_careers/education/standards/standards_glossary.html

[2] Young, P., Chaki, N., Berzins, V. and Luqi, Evaluation of middleware architectures in achieving system interoperability, Rapid Systems Prototyping Proceedings, 14th IEEE International Workshop 2003; p. 108-116

[3] ISO/TR 20514:2005, Health informatics -- Electronic health record -- Definition, scope and context by ISO/TC 215, Multiple. Distributed through American National Standards Institute, 2007; p. 1-27

[4] Hovenga, E.J.S., Health Information Governance in a Digital Environment, Grain, H (eds.), IoS Press, 2013; 193: p. 1-384

[5] Rules and Regulations, Department of Health and Human Services, Office of the Secretary, 45 CFR Parts 160 and 164, Modifications to the HIPAA Rules, Final Rule 2013, Federal Register, 2013;78: p. 1-17

[6] NIST, Guide for Mapping Types of Information and Information Systems to Security Categories, National Institute of Standards and Technology (NIST), NIST Special Publication 800-122, 2010; p. 1-59

[7] World Health Organization, Electronic Health Records: Manual for Developing Countries, 2006.

[8] Perumal T., Ramli A.R., Leong C.Y., Mansor S. and Samsudin K., Interoperability for Smart Home Environment Using Web Services.International Journal of Smart Home. 2008;2(4).

[9] Bhartiya, S. and Mehrotra, D., Threats and Challenges to Security of Electronic Health Records. In Proc. of QSHINE 2013, Social Informatics and Telecommunications Engineering, LNICST, 2013; 115: p. 543–559

[10] George, A.T. and Michael P. Interoperability among Heterogeneous Services. International Journal of Web Services Research, Zhang LiangJie, T.J. IBM eds., 2008; 5(4): p. 79-110.

[11] Dolin, R. H. and Alschuler, L., Approaching semantic interoperability in Health Level Seven, J Am Med Inform Assoc., 2011;18(1): p.99-103 12. Mao, M., Ontology Mapping: Towards Semantic Interoperability in Distributed and Heterogeneous Environments, ProQuest, 2008; p.1-163

[12] Melnik, S., and Decker S., A Layered Approach to Information Modeling and Interoperability on the Web, In Proceeding of the ECDL'00 Workshop on the Semantics Web, 2000. Retrieved from http://www-db.stanford.edu/~melnik/pub/sw0

[13] Carmagnola, F., Cena, F.,: From interoperable user models to interoperable user modeling, Proceedings of the 4th international conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Dublin, Ireland, 2006

[14] Bhartiya S. and Mehrotra D.: An Access Control Framework for Secured Sharing of Electronic Health Records using Hierarchy Similarity Analyzer, Int. J. of Electronic Healthcare, Inderscience, 2015;8(1): p. 25-50

[15] Hwang, KH, Chung, Kyo-IL, Chung, M-Ae., Choi, D., Review of Semantically Interoperable Electronic Health Records for Ubiqu-103itousHealthcare, Healthc Inform Res. 2010; 16(1): 1–5.

[16] Harvey,F., Kuhn,W., Pundt, H., Bishr, Y., Riedemann, C. Semantic interoperability: A central issue for sharing geographic information, The Annals of Regional Science, 1999; 33(2): p. 213-23

[17] Kalfoglou, Y., Schorlemmer, M., IF-Map: An Ontology-Mapping Method Based on Information-Flow Theory, S. Spaccapietra et al. (Eds.):

[18] Journal on Data Semantics, LNCS 2800, Springer-Verlag Berlin Heidelberg 2003; p. 98-127

APPENDIX A

List of Figures

List of Tables

List of Abbreviations

EHR	Electronic Health Record
GA	Genetic Algorithm
PCA	Principal Component Analysis

ANN	Artificial Neural Network
OTP	One Time Pad
OTT	One Time Tape

Symbols and Notations